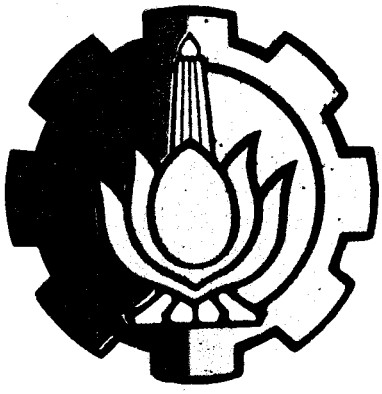
PERENCANAAN DAN PEMBUATAN PROTOTYPE SINKRONISASI TRAFFIC LIGHT DENGAN MENGGUNAKAN MIKROKOMPUTER 8031



OLEH :

FELIX PHANDEAN

NRP : 2882200986

JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
S U R A B A Y A

1993

PERENCANAAN DAN PEMBUATAN PROTOTYPE SINKRONISASI TRAFFIC LIGHT DENGAN MENGGUNAKAN MIKROKOMPUTER 8031

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar
Sarjana Teknik Elektro
Pada
Bidang Studi Elektronika
Jurusan Teknik Elektro
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
S u r a b a y a

Mengetahui / Menyetujui
Dosen Pembimbing

Ir. MOCH. HEROE

SURABAYA
PEBRUARI, 1993

A B S T R A K

Sinkronisasi traffic light berarti membuat sinkron atau selaras beberapa buah *traffic light* yang searah pada jarak tertentu. Alat ini dibuat dengan tujuan untuk mengurangi masalah kemacetan lalu lintas pada jalur prioritas. Pada perencanaan dan pembuatan *sinkronisasi traffic light* ini dibuat untuk tiga buah persimpangan, maka itu dapat diatur sedemikian rupa sehingga bila kendaraan yang mendapat prioritas lampu hijau pada *traffic light* yang pertama di perempatan pertama, juga akan mendapat lampu hijau pada *traffic light* di perempatan kedua dan ketiga. Dimana ketiga buah *traffic light* tersebut berada dalam arah jalur prioritas.

Perencanaan dan pembuatan *sinkronisasi traffic liglht* menggunakan komponen utama, yaitu mikrokomputer 8031 dan komponen penunjang seperti RAM 6116 yang berkapasitas 2 Kilobyte , ROM 7264 yang berkapasitas 8 Kilobyte (tempat program utama), dan PPI 8255 sebagai output ke lampu-lampu *traffic light* serta *display* yang menyatakan jam dalam sehari.

K A T A P E N G A N T A R

Atas Berkat Rahmat ALLAH Yang Maha Kuasa dan Maha Kasih, maka penulis berhasil menyelesaikan Tugas Akhir yang berjudul:

PERENCANAAN DAN PEMBUATAN PROTOTYPE SINKRONISASI TRAFFIC LIGHT DENGAN MENGGUNAKAN IC MIKROKOMPUTER 8031

Tugas Akhir ini merupakan salah satu syarat yang harus ditempuh untuk meraih gelar kesarjanaan di jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember, Surabaya.

Dalam mengerjakan Tugas Akhir ini penulis banyak mendapat bantuan dan bimbingan dari berbagai pihak. Pada kesempatan ini penulis menyampaikan ucapoan terima kasih kepada:

- Ir. Moch. Heroe, selaku dosen pembimbing dan dosen wali yang telah banyak memberikan bimbingan dan pengarahan kepada penulis dalam perencanaan dan pembuatan alat serta penyusunan naskah Tugas Akhir ini.
- Ir. Soetikno, selaku Koordinator Bidang Studi Elektronika Jurusan Teknik Elektro, Fakultas Teknologi Industri ITS yang telah memberikan persetujuannya kepada penulis untuk melaksanakan Tugas Akhir ini.
- Ir. Katjuk Astrowulan M.S.E.E., selaku ketua Jurusan Teknik Elektro , Fakultas Teknologi Industri ITS, yang

telah memberikan persetujuannya kepada penulis untuk melaksanakan Tugas Akhir ini.

- Seluruh staf pengajar dan administrasi Jurusan Teknik Elektro FTI ITS, yang telah membantu kelancaran pelaksanaan Tugas Akhir ini.
- Seluruh rekan-rekan mahasiswa Jurusan Teknik Elektro FTI ITS dan semua pihak yang telah turut membantu baik secara langsung maupun tidak langsung.

Akhir kata, penulis berharap semoga segala sesuatu yang telah dihasilkan dalam pembuatan Tugas Akhir ini dapat bermanfaat bagi kemajuan ilmu pengetahuan dan kesejahteraan umat manusia.

Surabaya, Pebruari 1993

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I

PENDAHULUAN

1.1. LATAR BELAKANG

Dengan bertambahnya jumlah penduduk di Indonesia maka akan diikuti dengan jumlah kendaraan yang semakin meningkat di jalan raya. Ini disebabkan karena kendaraan merupakan salah satu transportasi penduduk. Pesatnya jumlah kendaraan tidak seimbang dengan bertambahnya volume jalan, ini tentu menimbulkan kemacetan lalu lintas, kemacetan lalu lintas ini banyak terjadi terutama pada perempatan jalan, pada pusat keramaian kota, pada dua atau tiga atau lebih *traffic light* yang searah (satu jalur) dengan jarak tertentu. Tiga buah *traffic light* yang searah di Surabaya contohnya: jalan Genteng Kali, jalan Praban, dan jalan Bubutan atau jalan Darmawangsa/Airlangga, jalan Darmawangsa, dan jalan Ngagel Jaya, ataupun jalan Bengawan, jalan Kutai, dan jalan Adityawarman. Ketiga jalan tersebut sering terjadi kemacetan (ini merupakan suatu kasus atau masalah yang harus diatasi atau dipecahkan, ketiga jalan itu nantinya disebut sebagai jalur prioritas), oleh karena itu salah satu cara untuk mengatasi kemacetan lalu lintas adalah mengatur alur lalu lintas di setiap perempatan serta mensinkronkan antara ketiga jalan tersebut di atas.

Pengaturan lalu lintas bisa dilakukan dengan beberapa cara seperti: pemasangan rambu-rambu lalu lintas, bisa

juga dengan menggunakan tenaga manusia dalam hal ini adalah polisi lalu lintas, atau dengan alat kontrol, yaitu lampu lalu lintas (*traffic light*).

1.2. PERMASALAHAN

Masalah kemacetan pada dua atau tiga buah *traffic light* yang searah seperti yang telah disebutkan di atas merupakan suatu masalah yang harus diatasi agar kemacetan lalu lintas dapat dikurangi. Oleh karena itu dibuat suatu *traffic control* yang mana lama penyalaan lampu lalu lintas setiap phase dapat diprogram untuk setiap jam. Tugas akhir yang dibuat ini adalah untuk mensinkronkan ketiga buah *traffic light* yang berada pada jalur prioritas. Maksudnya adalah apabila kendaraan mendapat prioritas lampu hijau pada *traffic light* yang pertama, maka kendaraan tersebut ketika tiba di *traffic light* yang kedua dan ketiga juga akan mendapat lampu hijau, di mana ketiga buah *traffic light* tersebut berada pada satu jalur, yaitu jalur prioritas. Kendaraan yang disebutkan di atas adalah kendaraan apa saja (mobil dan sepeda motor) asal dapat mencapai kecepatan stabil 40 km/jam (kecepatan kendaraan menurut peraturan lalu lintas di dalam kota). Dalam hal ini memang jalur prioritas diutamakan tetapi juga tidak mengabaikan pengaturan lamanya penyalaan lampu-lampu lalu lintas lainnya di perempatan pertama, kedua, dan ketiga.

Kepadatan lalu lintas pada suatu jalan atau

perempatan tidak sama untuk setiap saat. Biasanya kepadatan lalu lintas terjadi pada keberangkatan dan kepulangan jam-jam kerja atau jam-jam sekolah. *Traffic control* tersebut dilakukan secara *hardware* maupun secara *software* untuk mendapatkan hasil optimum. Secara *hardware* menggunakan mikrokomputer 8031 sebagai komponen utama, dan komponen penunjang seperti ROM 2764, RAM 6116, PPI 8255 dan IC - IC TTL. Secara *software* menggunakan *software* mikrokomputer itu sendiri untuk menjalankan *hardware* dan bahasa Pascal untuk tampilan dan transfer data yang berupa lama penyalaan lampu-lampu lalu lintas.

1.3. TUJUAN

Tujuan dari Tugas Akhir yang berupa perencanaan dan pembuatan *prototype* sinkronisasi *traffic light* dengan menggunakan mikrokomputer 8031 adalah untuk mengurangi kemacetan lalu lintas terutama pada tiga buah *traffic light* yang berada pada satu jalur (jalur prioritas) sesuai dengan jarak yang diinginkan sehingga lalu lintas dapat berjalan dengan lancar dan tertib.

1.4. METODOLOGI

Pertama yang dilakukan adalah studi literatur mengenai segala sesuatu yang berhubungan dengan Tugas Akhir ini, terutama mempelajari fungsi dan cara kerja mikrokomputer 8031 beserta rangkaian penunjangnya.

Dalam perencanaan alat dilakukan perencanaan secara *hardware* dan secara *software*. Perencanaan *hardware* dengan mempelajari dan memahami IC mikrokomputer 8031, EPROM 2764, RAM 6116, PPI 8255, dan IC - IC pendukung lainnya. Perencanaan *software* dengan mempelajari fasilitas *software* mikrokomputer 8031 dalam menjalankan *hardware* dan kemampuannya mengatur input - output. Selain itu juga mempelajari *software* Pascal dalam mengirimkan data dari komputer ke mikrokontroler 8031. Sesudahnya dilakukan pembuatan peralatan baik secara hardware maupun secara software. Kemudian dilakukan pengujian dan pengukuran peralatan. Setelah semua berakhir maka dilakukan penulisan naskah Tugas Akhir.

1.5. SISTEMATIKA

Sistematika pembahasan pada tugas akhir ini dibagi dalam beberapa bab sebagai berikut:

- Bab I : Pendahuluan, bab ini membahas tentang latar belakang, permasalahan, pemecahan masalah, tujuan, metodologi, dan sistematika pembahasan.
- Bab II : Teori penunjang, bab ini membahas tentang penggunaan IC mikrokontroler 8031 sebagai komponen utama, dan EPROM 2764, RAM 6116, PPI 8255, serta IC - IC pendukung lainnya yang berfungsi sebagai komponen penunjang.

- Bab III : Perencanaan, bab ini akan membahas tentang teknik perencanaan perangkat keras dan perencanaan perangkat lunak.
- Bab IV : Pengukuran alat, bab ini membahas tentang pengujian unjuk kerja dan pengukuran lama nyalanya lampu-lampu lalu lintas dari peralatan yang dibuat.
- Bab V : Penutup, bab ini berisi kesimpulan dan saran dari tugas akhir ini.

BAB II

TEORI PENUNJANG

2.1. MIKROKONTROLER 8031[1)]

Mikrokontroler 8031 merupakan salah satu anggota keluarga MCS-51. Anggota keluarga MCS-51 lainnya adalah mikrokontroler 8051 dan mikrokontroler 8751. Mikrokontroler-mikrokontroler MCS-51 memiliki 40 pin (kaki) dan mikrokontroler 8031, 8051, 8751 tersebut memiliki konfigurasi pin-pin, pewaktuan (*timing*) dan karakteristik listrik yang sama. Perbedaan utamanya adalah dalam hal memori program *internal*-nya. Mikrokontroler 8751 memiliki 4 kilobyte EPROM (*Erasable Programmable Read Only Memory*). Mikrokontroler 8051 memiliki 4 kilobyte ROM (*Read Only Memory*) yang telah diisi program (sesuai dengan kehendak pemakai). Sedangkan mikrokontroler 8031 tidak memiliki memori program dalam, makanya mikrokontroler 8031 tersebut hanya menggunakan memori program *eksternal* (memori program luar).

Keluarga MCS-51 dalam kerjanya bisa mengakses 64 kilobyte memori program *eksternal* dan 64 kilobyte memori data *eksternal*. MCS-51 juga mempunyai 32 jalur input/output (I/O) dan *receive- buffered*, serial I/O dua arah. Dalam tugas akhir ini yang digunakan adalah mikrokontroler 8031 maka hanya segala sesuatu yang

---

1). Volume II, Intel Corp., Santa Clara California USA, 1986, hal. 7-1 s.d. 7-2

mengenai mikrokontroler 8031 saja yang akan dibahas.

2.1.1. ARSITEKTUR 8031[2)]

Blok diagram struktur *internal* dari mikrokontroler 8031 ditunjukkan pada gambar 2.1. Secara umum struktur atau organisasi *internal* mikrokontroler 8031 adalah sebagai berikut:

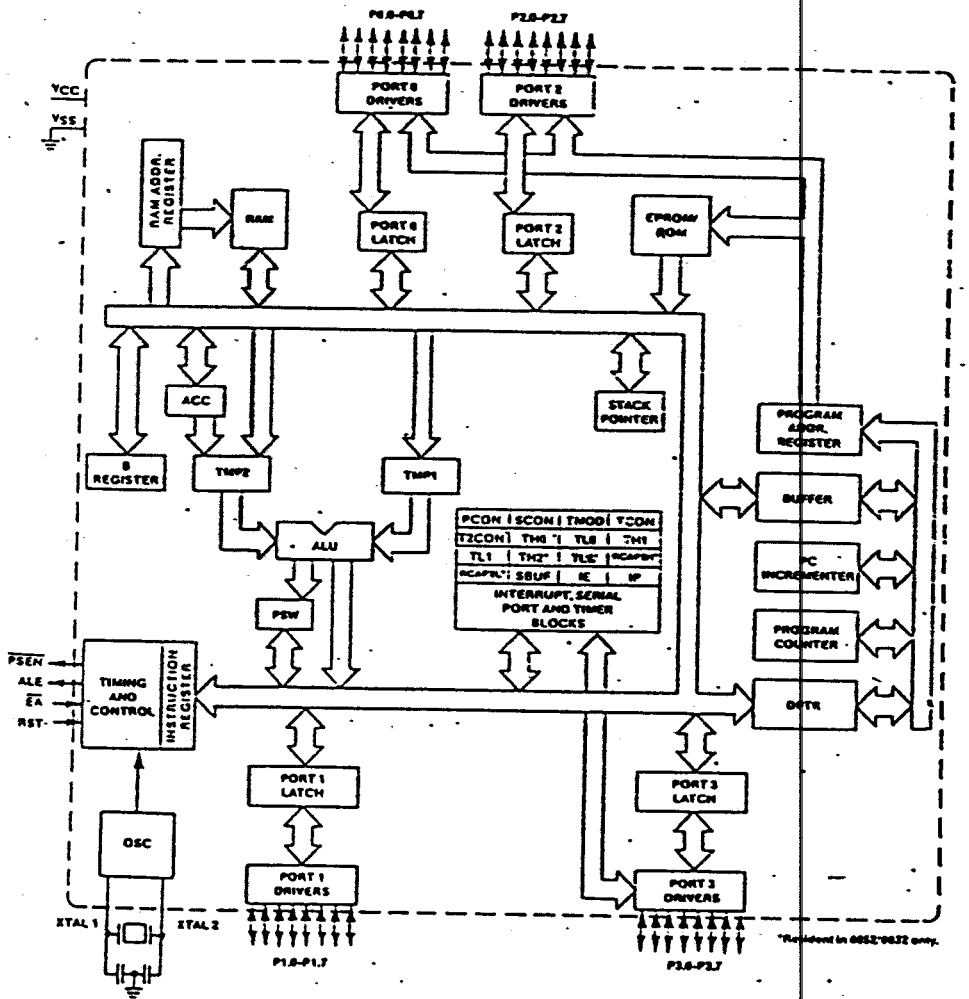
- CPU 8 bit
- RAM 128 byte
- 21 register fungsi khusus
- Kedalaman stack hanya dibatasi oleh RAM data *internal*
- 32 I/O *line*
- 64K ruang alamat untuk memori data luar
- 64K ruang alamat untuk memori program luar
- Osilator
- Dua *timer/counter* 16 bit
- Lima *interupt* dengan struktur dua tingkat prioritas
- Serial I/O *full-duplex*
- Kemampuan pengalamatan bit untuk proses *boolean*.

Memori data *internal* kemudian terbagi menjadi dua, yaitu 128 byte RAM data internal dan 128 byte register fungsi khusus (SFR = *Special Function Registers*). RAM data *internal* mempunyai empat buah register Bank di mana

---

2). Wharthon, John, An Introduction to the Intel MCS-51 Single-Chip Microcomputer Family, Intel Application Note (AP-69), Intel Corp., Santa Clara USA, 1980, hal. 4 s.d 13

setiap bank terdiri dari delapan buah register, 128 bit yang bisa dialamati (128 *addresable bit*), dan stack. Lokasi stack ditentukan oleh penunjuk stack (*Stack Pointer*) 8 bit.



Gambar 2.1. [3)]

Blok diagram struktur *internal* 8031

3). Ibid, hal. 5

Semua register, kecuali *Program Counter* dan kedelapan buah Register Bank terdapat dalam ruang alamat Register Fungsi Khusus (SFR). Register-register tersebut adalah Register Aritmatika, Penunjuk (*pointer*), I/O port, Register-register untuk sistem *interupt*, *timer* dan saluran serial. Lokasi 128 bit pada SFR dapat dialamati sebagai bit.

Pada umumnya instruksi-instruksi 8031 memiliki satu byte, yaitu 49 instruksi *single byte*, 45 instruksi *two byte*, dan 17 instruksi *three-byte*. Jika menggunakan kristal 12 MHz, 64 instruksi memiliki waktu eksekusi 1 us dan 45 instruksi memiliki waktu eksekusi 2 us, sedangkan 2 instruksi sisanya, yaitu perkalian dan pembagian memiliki waktu eksekusi 4 us.

2.1.2. FUNGSI PIN-PIN MIKROKONTROLER 8031[4)]

Konfigurasi dari pin-pin mikrokontroler 8031 ditunjukkan pada gambar 2.2. Fungsi tiap-tiap pinnya adalah sebagai berikut:

- Vss : Pin ini dihubungkan dengan *ground* dari sumber tegangan rangkaian.
- Vcc : Pin ini dihubungkan dengan sumber tegangan 5 volt.
- Port 0 : Port 0 merupakan port I/O 8 bit dua arah. Port

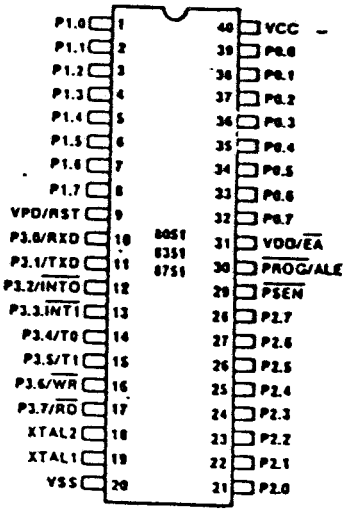
---

4). Volume II, op. cit., hal. 7-92 s.d. 7-93

ini juga digunakan sebagai *multiplex bus* alamat rendah dan bus data selama pengaksesan ke memori luar . Port 0 bisa dibebani dengan dua buah TTL

- Port 1 : Port 1 merupakan port I/O 8 bit dua arah quasi. Port 1 bisa dibebani dengan 1 buah TTL.
- Port 2 : Port 2 adalah port I/O 8 bit *quasi-bidirectional*, mengirimkan alamat tinggi ketika mengakses memori *eksternal*. Bisa dibebani dengan satu buah TTL
- Port 3 : Port 3 adalah port I/O 8 bit *quasi-bidirectional*, meliputi *interupt*, *timer*, serial port, pin $\overline{RD}$, dan $\overline{WR}$. Port 3 bisa dibebani dengan satu buah TTL. Port 3 juga memiliki fungsi-fungsi khusus sebagai berikut:
  - RXD/data (P3.0), port serial sebagai penerima masukan data *asynchronous* atau masukan/keluaran data *synchronous*.
  - TXD/clock (P3.1), port serial sebagai pengirim keluaran data *asynchronous* atau keluaran *clock synchronous*.
  - INT0 (P3.2), input *interupt* 0 atau gerbang input kontrol untuk *counter* 0.
  - INT1 (P3.3), input *interup* 1 atau gerbang input kontrol untuk *counter* 1.
  - T0 (P3.4), masukan untuk *counter* 0
  - T1 (P3.5), masukan untuk *counter* 1.

- WR (P3.6). Sinyal kontrol penulisan men-*latches* byte data dari port 0 ke memori data *eksternal*.
- RD (P3.7). Sinyal kontrol pembacaan meng-*enables* memori data *eksternal* ke port 0.



Gambar 2.2.[5]

Konfigurasi pin-pin 8031

- RST/Vpd : Perubahan tegangan dari rendah ke tinggi pada pin ini (kira-kira 3 V) akan mereset 8031.
- ALE/$\overline{\text{PROG}}$: ALE (*address Latch Enable*) menghasilkan output yang digunakan untuk mengunci alamat ke memori luar selama operasi normal.
- PSEN : Output PSEN (*Program Store Enable*) merupakan

5). Wharton, John, op. cit., hal. 1

sinyal kendali yang menghubungkan memori program *eksternal* dengan bus selama operasi normal.

- -EA/Vdd : Bagi mikrokontroler 8031, pin ini harus dihubungkan dengan *ground* agar bisa melaksanakan instruksi dari memori program *eksternal*.
- XTAL1 : Input ke *amplifier* osilator yang berpenguatan tinggi. Pin ini dihubungkan dengan kristal atau sumber osilator dari luar.
- XTAL2 : output *amplifier* osilator. Pin ini dihubungkan dengan kristal jika menggunakan sumber osilator dari dalam.

2.1.3. PERANGKAT KERAS CPU[6)]

Mikrokontroler 8031 terbagi dalam beberapa blok. Di mana fungsi dari tiap-tiap blok sebagai berikut:

- Dekoder instruksi : Dekoder instruksi inilah yang akan menerjemahkan setiap instruksi program serta dapat menghasilkan sinyal yang mengatur fungsi dari setiap bagian dalam CPU.
- Program *counter* : Program *counter* (PC) 16 bit berfungsi untuk mengontrol urutan instruksi yang akan dilaksanakan.
- Memori program dalam : 8051/8751 memiliki 4 kilobyte memori program yang terdapat dalam serpih (*chip*). Sedangkan 8031 tidak memiliki memori

6). Ibid., hal. 5 s.d. 8.

program di dalam serpih.

- RAM data dalam : Register-register yang terdapat pada 128 byte RAM data dalam adalah sebagai berikut:
  - Bank register, terdapat 4 buah bank register di dalam RAM data dalam, di mana setiap banknya mempunyai register R0 sampai R7.
  - 128 *Addressable bit*, terdapat dalam 16 byte yang berlokasi pada alamat 20H sampai 2FH di dalam RAM data dalam.
  - Stack, lokasi stack dapat mencapai 128 byte dan stack tersebut bisa ditempatkan di mana saja pada RAM data dalam.
- Register fungsi khusus (SFR) : Register-register yang terdapat di dalam register fungsi khusus adalah sebagai berikut:
  - Register A : Register ini digunakan sebagai akumulator.
  - Register B : Register ini digunakan bersama register A untuk instruksi perkalian dan pembagian.
  - *Register program status word* (PSW): *Carry* (CY), *auxiliary carry* (AC), *user flag* 0 (F0), pemilih bank register (RS0 dan RS1), *overflow* (OV) dan *parity flag* (P) terdapat dalam *register program status word*. *Flag* CY, AC, dan OV biasa digunakan untuk menyatakan kondisi dari operasi aritmatika yang terakhir. *Flag* P merupakan parity

dari register A. *Flag carry* juga digunakan sebagai akumulator *Boolean* untuk operasi bit.

- DPTR : Data *Pointer* (DPTR) adalah register 16 bit, yang terdiri dari prenunjuk data tinggi (data *pointer high*) dan penunjuk data rendah (data *pointer low*). Fungsinya adalah untuk memegang alamat 16 bit sebagai *pointer* ke ROM dan RAM luar.
- P0, P1, P2, dan P3 : Empat buah port tersebut mempunyai 32 jalur I/O untuk berhubungan dengan perangkat keras lainnya di luar. Semua port bisa dapat dialamati secara bit maupun byte. Port 0 (P0) dan port 2 (P2) bisa digunakan untuk menambah jumlah memori luar. Port 3 (P3) memiliki sinyal kontrol khusus seperti sinyal baca dan sinyal tulis. Port 1 (P1) hanya bisa digunakan untuk I/O.
- *Register Interupt Priority* (IPC) : *Register Prioritas Interupt* (IPC) memiliki bit-bit kontrol untuk mengaktifkan *interrupt* pada taraf yang diinginkan.
- *Register Interrupt Enable* (IEC) : IEC memiliki bit-bit kontrol untuk mengaktifkan kelima sumber *interrupt* dan untuk menghidupkan/mematikan (*enable/disable*) keseluruhan interupt.
- Register mode waktu/pencacah (*Timer/Counter Mode Register* / TMOD) : Bit-bit yang terdapat dalam register TMOD digunakan untuk memilih

pewaktu/pencacah yang akan bekerja.

- Register kontrol pewaktu/pencacah (*timer/counter control register* / TCON) : Semua pewaktu/pencacah dikontrol oleh bit-bit dari register TCON. Bit-bit mulai/berhenti (*start/stop*) untuk semua pewaktu/pencacah, *flag-flag overflow* dan permintaan *interupt* terdapat di dalam register TCON.
- Register-register pewaktu/pencacah 1 tinggi dan rendah (TH1 dan TL1), pewaktu/pencacah 0 tinggi dan rendah (TH0 dan TL0): Terdapat 4 lokasi register untuk 2 buah *timer* atau *counter* 16 bit. Register-register ini dapat dibaca dan ditulis. TH1 dan TH0 digunakan untuk 8 bit tinggi dari *timer/counter* 1 dan 0. TL1 dan TL0 digunakan untuk 8 bit rendah dari *timer/counter* 1 dan 0.

2.1.4. RANGKAIAN OSILATOR [7)]

Rangkaian osilator yang terdapat di dalam mikrokontroler 8031 adalah rangkaian pararel anti-resonansi dengan frekuensi antara 1,2 MHz sampai 12 MHz. Frekuensi tersebut masih dibagi 12 lagi oleh pewaktuan *internal* yang mana memberikan instruksi minimum 1 us dengan kristal 12 MHz pada mikrokontroler 8031.

---

7). Volume II, op. cit., hal. 7-9 s.d. 7-4.

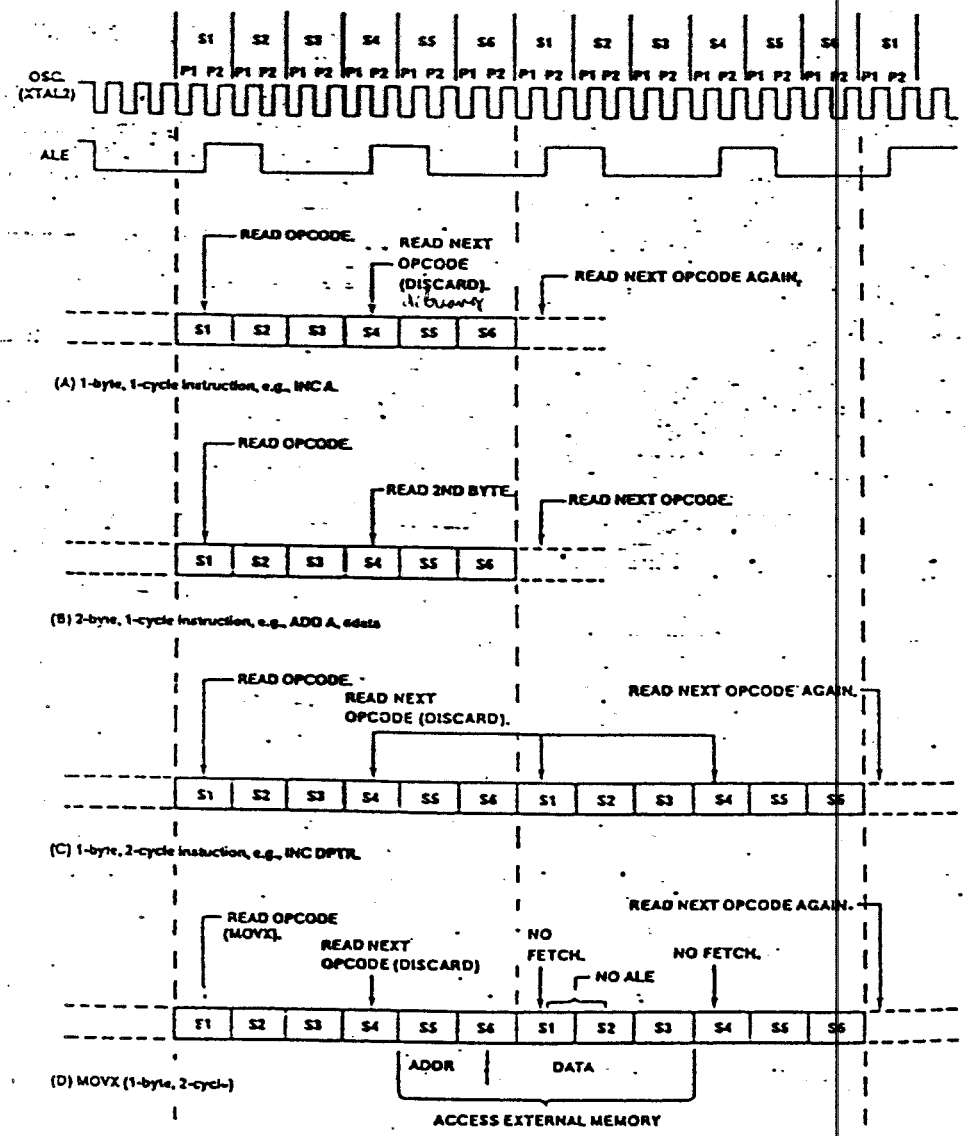
2.1.5. PEWAKTUAN CPU[8)]

Satu *machine cycle* terdiri dari 6 keadaan (12 perioda osilator) dan setiap keadaan terbagi dalam dua fase yang berhubungan dengan dua fase sinyal *clock*. Biasanya untuk operasi-operasi aritmatika dan logika dilaksanakan pada fase 1 dan transfer register ke register *internal* dilaksanakan pada fase 2. Diagram pewaktuan pengambilan/pelaksanaan instruksi yang berhubungan dengan keadaan dan fase *internal* diperlihatkan pada gambar 2.3. Karena sinyal *clock internal* ini tidak biasa diamati dari luar, maka sinyal XTAL2 dan ALE digunakan sebagai referensi *eksternal*. Satu *machine cycle* terdiri dari 12 perioda osilator, diberi nomor dari S1P1 (*State* 1 *Phase* 1) sampai S6P2 (*State* 6 *Phase* 2). Setiap *state* mempunyai waktu durasi selama dua perioda osilator dan setiap fase terakhir pada satu perioda osilator. ALE aktif dua kali setiap *machine cycle*, sekali selama S1P1 dan S2P1, dan sekali lagi selama S4P2 dan S5P1.

Eksekusi dari instruksi *one-cycle* mulai pada S1P2, ketika *opcode* dilewatkan ke dalam register instruksi. Bila mengeksekusi instruksi dua byte, byte yang kedua dibaca selama S4 dari *machine cycle* yang sama. Bila itu instruksi satu byte, tetap dibaca pada S4, tetapi pembacaan byte yang berupa *opcode* berikutnya diabaikan,

8). Ibid., hal. 7-3 s.d. 7-4.

dan programm *counter* tidak dinaikkan (not incremented). Pada umumnya, eksekusi selesai pada akhir dari S6P2.



Gambar 2.3. [9]

Diagram pewaktuan eksekusi instruksi

Gambar 2.3.(a) dan (b) menunjukkan diagram pewaktuan untuk instruksi 1 byte, 1 cycle dan untuk instruksi dua

9). Ibid., hal. 7-4

byte, 1 *cycle*. Kebanyakan instruksi-instruksi dieksekusi dalam satu *cycle*. Hanya instruksi perkalian dan pembagian yang memerlukan lebih dari dua *cycle* untuk menyelesaikannya, instruksi-instruksi tersebut memerlukan 4 *cycle*.

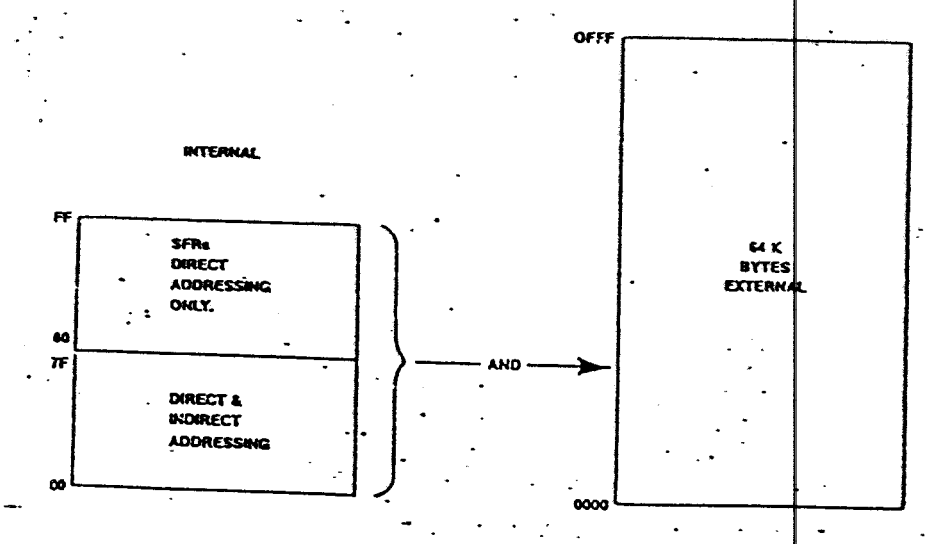
Selama setiap *machine cycle*, dua byte kode akan diambil dari memori program. Satu-satunya perkecualian adalah ketika mengeksekusi instruksi MOVX. MOVX adalah instruksi 1 byte, 2 cycle yang mengakses memori data *eksternal*, dimana selama data *eksternal* diakses dan di-*strobe* ada 2 kali pengambilan *opcode* yang diabaikan. Gambar 2.3(c) dan (d) menunjukkan diagram pewaktuan untuk instruksi 1 byte, 2 cycle dan untuk instruksi MOVX.

2.1.6. DATA MEMORY[10]

Memory data terdiri dari 128 byte dalam RAM internal, 21 *spesial function register* (register fungsi khusus) dan hingga 64K byte memori data *eksternal* seperti ditunjukkan pada gambar 2.4. Memory data *eksternal* bisa menggunakan baik *address* 8 bit ataupun 16 bit, yang menyediakan ruang *address* 256 lokasi. *Address* 128 byte terendah mengakses ke dalam RAM *internal*. Sedangkan lokasi pada 128 byte teratas ditempati spesial *function* register (SFR).

---

10). Ibid., hal. 8-3 s.d. 8-4.



Gambar 2.4[11]

Data memory mikrokontroler 8031

32 bit terendah dari *internal* RAM (lokasi 00H hingga 1FH) dibagi menjadi 4 bank register, yang masing-masingnya terdiri dari 8 byte. Pada umumnya sebagian besar instruksi hanya menggunakan satu byte. 16 byte berikutnya dari *internal* RAM (lokasi 20H hingga 2FH) memiliki bit-bit yang bisa di-*addres* secara terpisah. Di samping 128 bit yang bisa di-*address* terpisah di dalam RAM, sebelas spesial *function* register juga mempunyai bit-bit yang bisa di-*address* terpisah.

2.1.7. MENGAKSES MEMORY EKSTERNAL[12]

Mikrokontroler 8031 bisa mengakses 64 kilobyte memori

11). Ibid., hal. 8-4.

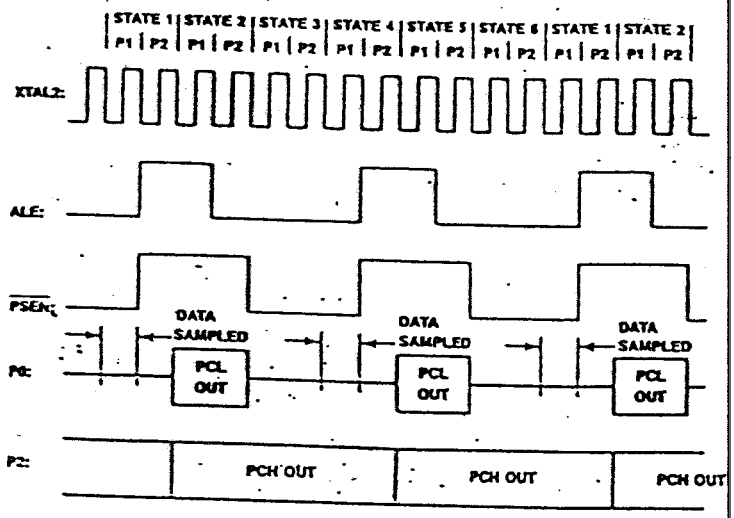
12). Ibid, hal 7-8 s.d. 7-9.

data eksternal dan 64 kilo byte memori program *eksternal*. Pengaksesan memori *eksternal* bisa dilaksanakan jika pin $\overline{EA}$ rendah. Sinyal ALE, $\overline{PSEN}$, $\overline{RD}$, dan $\overline{WR}$ digunakan sebagai sinyal kontrol memori. Sinyal ALE digunakan untuk menahan alamat ke memori *eksternal*. Sinyal $\overline{PSEN}$ (*Program Store Enable*) digunakan untuk mengakses memori program *eksternal*, sedangkan sinyal $\overline{RD}$ dan $\overline{WR}$ digunakan untuk mengakses memori data *eksternal*. Ketika mengakses memori *eksternal*, mikrokontroler 8031 melalui port 2 akan mengeluarkan byte alamat tinggi dan port 0 akan mengeluarkan byte alamat rendah/data.

Pada setiap siklus bus memori program terbagi atas enam perioda osilator. Setiap perioda dinamakan T1 sampai T6. Alamat dikeluarkan dari mikrokontroler selama T3. *Transfer* data terjadi pada bus selama T5, T6, dan siklus bus berikutnya T1. Siklus pembacaan mulai pada T2 dengan adanya sinyal ALE. Pada akhir dari sinyal ALE digunakan untuk melewatkan informasi alamat yang ada di bus saat itu. Pada T5, alamat dihilangkan dari bus port 0 dan pengendali bus berada pada keadaan impedansi tinggi. Kontrol pembacaan memori program akan berada pada T5, $\overline{PSEN}$ menyebabkan komponen yang dialamati meng-*enable* pengendali busnya, sehingga tidak lama kemudian data instruksi yang diinginkan telah berada pada bus. Ketika 8031 mengembalikan sinyal $\overline{PSEN}$ pada keadaan *high*,

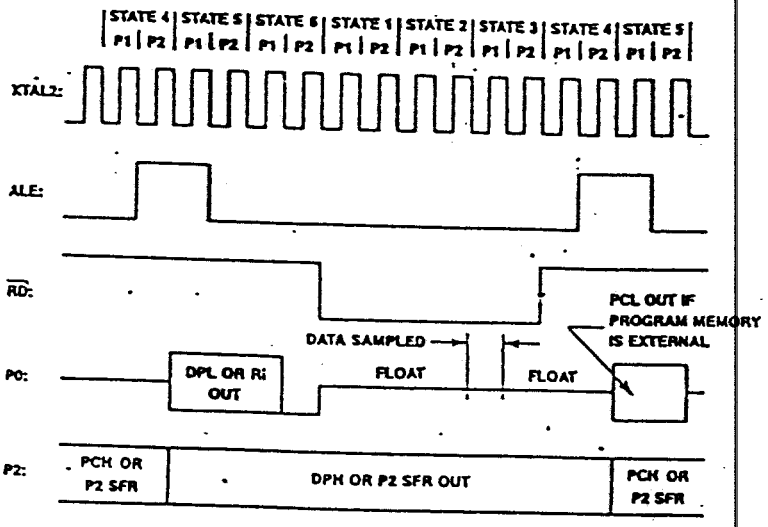
pengendali bus dari komponen yang dialamati akan kembali mengambang.

Pada setiap siklus bus memori data eksternal terbagi atas dua belas perioda osilator, yang dinamakan T1 sampai T12. Alamat dikeluarkan dari mikrokontroler selama T3. *Transfer* data terjadi pada bus selama T7 sampai T12. T5 dan T6 adalah perioda di mana arah dari bus diubah untuk operasi pembacaan. Siklus pembacaan dimulai pada T2 dengan adanya sinyal ALE. Pada akhir sinyal ALE digunakan untuk melewatkan informasi alamat yang telah ada pada bus pada saat ini. Pada T5, alamat dihilangkan dari bus dan bus port 0 berada pada keadaan impedansi tinggi. Sinyal kontrol pembacaan memori $\overline{RD}$, keluar selama T7. Sinyal $\overline{RD}$ menyebabkan komponen yang dialamati meng-*enable* pengendali busnya. Tidak lama kemudian data yang diinginkan akan berada pada bus. Ketika 8031 mengembalikan sinyal $\overline{RD}$ pada keadaan *high*, pengendali bus dari komponen yang dialamati akan mengambang. Siklus penulisan dimulai dengan sinyal ALE dan pengiriman alamat. Pada T6, mikrokontroler mengirimkan data untuk ditulis ke dalam lokasi data memori yang dialamti. Data ini tetap ada pada bus sampai akhir siklus bus berikutnya T2. Sinyal tulis $\overline{WR}$ menjadi *low* pada T6 dan tetap aktif sampai T12.



Gambar 2.5.[13)]

Siklus waktu pembacaan memori program



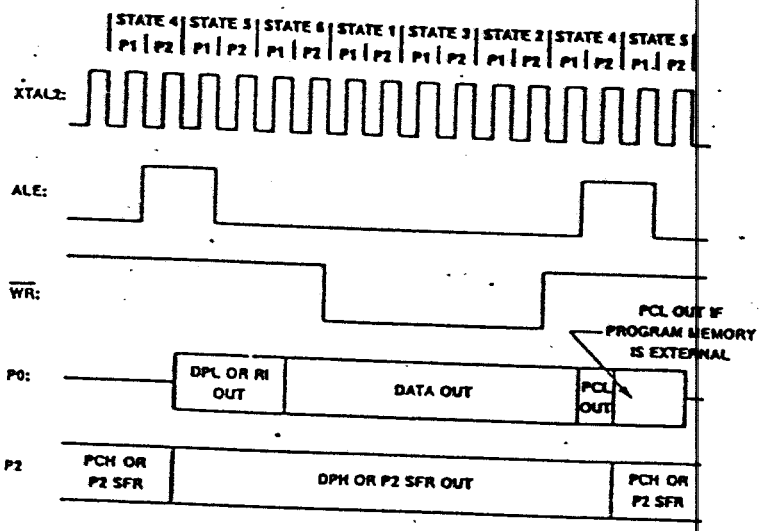
Gambar 2.6.[14)]

Siklus waktu pembacaan memori data

---

13). Ibid., hal. 7-34.

14). Ibid., hal. 7-34.



Gambar 2.7.[15]

Siklus waktu penulisan memori data

Pada Gambar 2.5, gambar 2.6, gambar 2.7 secara berurutan menunjukkan blok diagram pewaktuan dari proses pembacaan memori program, pembacaan memori data dan penulisan memori data.

2.1.8. SISTEM INTERRUPT[16]

Mikrokontroler 8031 memiliki lima sumber *interrupt* yang masing-masing bisa diprogram pada salah satu dari dua tingkat prioritas. Kelima sumber interrupt itu adalah :

- INTO :

15). Ibid., hal. 7-35.

16). Ibid., hal. 7-23 s.d. 7-25.

Permintaan luar (*eksternal request*) dari kaki P3.2.

- INT1 :

  Permintaan luar dari kaki P3.3.

- *Timer* 0 :

  Overflow dari *timer* 0 akan mengaktifkan flag permintaan *interrupt* (*interrupt request flag*) TF0.

- *Timer* 1 :

  Overflow dari *timer* 1 akan mengaktifkan flag permintaan interrupt TF1.

- Port Serial :

  Flag T1 akan aktif jika pengiriman satu *frame* serial telah lengkap dan flag RI akan aktif jika penerimaan satu *frame* serial telah lengkap.

Tabel 2.1.[17)]

Alamat awal Interrupt Service Routine

Sumber Interrupt	Alamat awal
Interrupt External 0 (INT0)	0003H
Timer/counter 0 (T0)	000BH
Interrupt External 1 (INT1)	0013H
Timer/counter 1 (T1)	001BH
Port Serial	0023H

Jika ada permintaan interrupsi dari luar maka Program *Counter* akan meloncat pada alamat-alamat awal dari

17). Ibid, hal. 7-25.

Interrupt Service Program, seperti tampak pada tabel 2.1. Setiap sumber *interrupt* dapat di-*enable* atau di-*disable* secara tersendiri dengan mengubah bit *set* atau *clear* pada register *Interrupt Enable* (IE). Semua sumber *interrupt* juga dapat di-*enable* atau di-*disable* secara keseluruhan.

Sedangkan prioritas *interrupt* dipilih dengan mengubah bit-bit dari register *Interrupt Priority* (IP). Pada gambar 2.8 ditunjukkan bit-bit dari register EI. Sedangkan pada gambar 2.9 ditunjukkan bit-bit dari register IP. Bit- bit dari register IE adalah sebagai berikut:

- EA : *Enable All*. Jika EA = 0, semua interrupt di-*disable*. Sedangkan bila EA = 1, masing-masing *interrupt* sumber *enable* atau *disable*, di-*set* atau *clear* oleh *enable* bit ini.
- ES : *Enable* atau *disable interrupt* serial port. Bila ES = 0, *interrupt* port serial di-*disable*.

(MSB) (LSB)

7	6	5	4	3	2	1	0
EA	X	X	ES	ET1	EX1	ETO	EXO

Gambar 2.8.[18]

Register Interrupt Enable

- ET1 : *Enable Timer* 1. Bila ET1 = 0, *interrupt timer* 1

18). Ibid, hal. 8-13

di-*disable*.

- EX1 : *Enable External Interrupt* 1. Bila EX1 = 0, *eksternal interrupt* 1 di-*disable*.
- ET0 : *Enable Timer* 0. Bila ET0 = 0, *interrupt timer* 0 di-*disable*.
- EX0 : *Enable External Interrupt* 0. Bila EX0 = 0, *eksternal interrupt* 0 di-*disable*.

Dengan mengubah bit-bit yang bersesuaian pada register IP maka setiap sumber *interrupt* dapat diprogram menjadi tingkat prioritas tinggi atau rendah sesuai dengan yang diinginkan. *Interrupt* prioritas rendah bisa diinterupsi oleh *interrupt* prioritas tinggi, tetapi tidak bisa diinterrupsi oleh *interrupt* prioritas rendah yang lain. Sedangkan *interrupt* prioritas tinggi tidak bisa diinterrupsi oleh *interrupt* prioritas rendah. Bit-bit yang terdapat dalam register IP adalah sebagai berikut:

- PS : Serial Port Priority level. Bila PS = 1, tingkat prioritasnya menjadi lebih tinggi.
- PT1 : *Timer* 1 priority level. Bila PT1 = 1, tingkat prioritasnya menjadi lebih tinggi.
- PX1 : *Eksternal Interrupt* 1 Priority level. Bila PX1 = 1, tingkat priritasnya menjadi lebih tinggi.
- PT0 : *Timer* 0 priority level. Bila PT0 = 1, tingkat prioritasnya menjadi lebih tinggi.
- PX0 : *eksternal interrupt* 0 priority level. Bila PX0 = 1, tingkat prioritasnya menjadi lebih tinggi.

(MSB) (LSB)

7	6	5	4	3	2	1	0
X	X	X	PS	PT1	PX1	PT0	PX0

Gambar 2.9.[19]

Register Interrupt Priority

(MSB) (LSB)

7	6	5	4	3	2	1	0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

Gambar 2.10.[20]

Register TCON

Register TCON digunakan untuk menentukan trigger dari *eksternal interrupt*; sisi jatuh (*falling edge*) atau keadaan rendah (*low level*); menjalankan atau menghentikan *timer/counter*. *Timer Overflow Flag* pada register ini akan di-*set* oleh perangkat keras bila *timer/counter overflow* dan *Interrupt Edge Flag* di-*set* bila terdeteksi perubahan dari *logic* '1' ke *logic* '0' dari sinyal *interrupt* luar. Pada gambar 2.10 menunjukkan susunan bit-bit dari register TCON.

19). Ibid., hal. 8-14

20). Ibid., hal. 8-15.

2.1.9. TIMER/COUNTER[21)]

Mikrokontroler 8031 memiliki dua buah register 16 bit yang dapat digunakan sebagai pewaktu/pencacah. Register-register tersebut adalah TH0 (pewaktu/pencacah 0 byte tinggi), TL0 (pewaktu/pencacah 0 byte rendah), TH1 (pewaktu/pencacah 1 byte tinggi) dan TL1 (pewaktu/pencacah 1 byte rendah). Setiap pewaktu/pencacah dikontrol oleh bit dalam register TMOD untuk memilih fungsi sebagai pewaktu atau pencacah. Terdapat empat buah mode pewaktu/pencacah yang dapat dipilih, yaitu:

- Mode 0 :

  Menghasilkan pencacah/pewaktu 8 bit dengan *prescaler* pembagi 32. Register TH1 atau TH0 sebagai pewaktu/pencacah, register TL1 atau TL0 bit 4-0 sebagai *prescaler*, sedangkan bit 5-7 diabaikan.

- Mode 1 :

  Menghasilkan pewaktu/pencacah 16 bit. TH1 dan TL1 atau TH0 dan TL0 di-*kaskade*.

- Mode 2 :

  Menghasilkan pewaktu/pencacah 8 bit yang *autoreload*. Register TH1 atau TH0 berisi bilangan yang akan di-*reload* ke register TL1 atau TL0 setiap kali *overflow*.

- Mode 3 :

  Pewaktu 0 dibagi menjadi dua buah pewaktu/pencacah 8 bit yang terpisah. Pewaktu 1 tidak berjalan. Gambar 2.11

---

21). Ibid., hal. 7-10 s.d. 7-12.

2.1.10. PORT SERIAL[23]

Salah satu keunggulan mikrokontroller 8031 adalah dengan adanya port serial di dalam serpih. Port serial ini bisa digunakan untuk komunikasi data serial, baik yang menggunakan hubungan half duplex maupun full duplex, atau untuk menambah port I/O dengan menghubungkan ke register geser (*shift register*). Port serial ini bisa dioperasikan dalam empat mode, yaitu:

- Mode 0 (*synchronous*)

  Data serial 8 bit dikirim dan diterima melalui RXD, dengan bit terendah (LSB) yang pertama, dan TXD mengeluarkan clock penggeser (*shift clock*). Baud rate tetap yaitu 1/12 dari frekuensi osilator.

- Mode 1 (*asynchronous*)

  Data serial 10 bit dikirimkan melalui TXD atau diterima melalui RXD dengan urutan: satu start bit, 8 bit data (LSB yang pertama) dan satu stop bit. Pada penerimaan data, stop bit akan mengisi RB8 pada register SCON. Baud rate dapat diubah-ubah (*variable*).

- MOde 2 (*aynchronous*)

  Data serial 11 bit dikirimkan melalui TXD atau diterima melalui RXD dengan urutan: satu start bit, 8 bit data (LSB yang pertama), satu bit data yang dapat diprogram (bit ke sembilan), dan satu stop bit. Pada pengiriman data, bit data ke sembilan (TB8) dapat dipilih '0' atau

---

23). Ibid., hal. 7-13 s.d. 7-14.

'1'. TB8 dapat digunakan sebagai parity bit. Pada penerimaan data, bit data ke sembilan akan mengisi RB8 pada register SCON, dan stop bit diabaikan. Baud rate dapat diprogram pada 1/32 atau 1/64 dari frekuensi osilator.

- Mode 3 (*asynchronous*)

  Mode 3 sama dengan mode 2 kecuali pada mode 3 baud rate dapat diubah-ubah (*variable*).

2.1.11. BAUD RATE [24]

Baud rate pada mode 0 adalah tetap, yaitu 1/12 dari frekuensi osilator. Baud rate pad mode 2 adalah 1/64 atau 1/32 dari frekuensi osilator, tergantung dari bit SMOD pada register PCON. Bila SMOD = 0, baud rate menjadi 1/64 dari frekuensi osilator, sedangkan bila SMOD = 1, baud rate menjadi 1/32 dari frekuensi osilator.

Sedangkan Baud rate pada mode 1 dan 3 ditentukan dari kecepatan *overflow* pewaktu 1, dengan persamaan:

Baud rate = kecepatan overflow pewaktu 1 / n

Nilai dari bilangan n ditentukan oleh bit SMOD. Bila SMOD = 0, n = 32 dan bila SMOD = 1, n = 16. Kecepatan *overflow* pewaktu 1 ditentukan oleh kecepatan pencacahan dan banyaknya pencacahan yang diperlukan untuk mencapai overflow. Persamaan untuk kecepatan *overflow* adalah:

---

24). Ibid., hal. 7-15 s.d. 7-17.

Kecepatan *overflow* = kecepatan pencacahan / (256 - TH1)

TH1 adalah nilai reload yang diisikan dengan perangkat lunak pada register TH1.

Tabel 2.2.[25)]

Penggunaan baud rate

BAUD RATE	FREKUENSI OSILATOR	SMOD	TIMER 1		
			C/$\overline{T}$	MODE	NILAI RELOAD
Mode 0 max: 1 MHz	12 MHz	x	x	x	x
Mode2 max: 375KHz	12 MHz	x	x	x	x
Mode 1&3 : 62 KHz	12 MHz	1	0	2	FFH
19,2 KHz	11,059 MHz	1	0	2	FDH
9,6 KHz	11,059 MHz	0	0	2	FDH
4,8 KHZ	11,059 MHz	0	0	2	FAH
2,4 KHz	11,059 MHz	0	0	2	F4H
1,2 KHz	11,059 MHz	0	0	2	E8H
137,5 Hz	11,059 MHz	0	0	2	1DH
110 Hz	8 MHz	0	0	2	72H
110 Hz	12 MHz	0	0	1	FEEBH

Baud rate yang biasa digunakan, ditunjukkan pada tabel 2.2. Dalam tugas akhir ini digunakan serial port

25). Ibid., hal. 7-16.

mode 1 dengan baud rate sebesar 4,8 KHz. Sehingga nilai TH1 diisikan dengan nilai 243, sesuai dengan perhitungan dari rumus di atas.

2.1.12. REGISTER SERIAL PORT CONTROL (SCON)[26]

Untuk mendefinisikan mode operasi dan kontrol fungsi-fungsi tertentu dari port serial digunakan bit-bit dalam register SCON. Register SCON ditunjukkan pada gambar 2.12 seperti di bawah ini. SM0 dan SM1 digunakan untuk memilih mode operasi dari port serial seperti yang ditunjukkan pada tabel 2.3. Sedangkan fungsi bit-bit yang lain adalah sebagai berikut:

- SM2

  Pada mode 2 atau 3, bila SM2 = 1, maka RI tidak akan diaktifkan bila bit data ke 9 (RB8) yang diterima adalah '0'. Pada mode 1, bila SM2 = 1, maka RI tidak akan diaktifkan bila stop bit yang sempurna tidak diterima. Pada mode 0 SM2 tidak digunakan.

- REN

  Untuk meng-*enable/disable* penerimaan data, maka bit REN di-*set/clear* oleh perangkat lunak.

- TB8

  Bit data ke 9 yang akan dikirimkan pada mode 2 dan 3.

- RB8

  Pada mode 2 dan 3, RB8 sebagai bit data ke 9 yang

---

26). Ibid., hal. 8-22.

diterima. Pada mode 1 bila SM2 = 0, RB8 adalah stop bit yang diterima. Pada mode 0 RB8 tidak digunakan.

Tabel 2.3.[27]

Pemilihan mode operasi port serial

SMO	SM1	MODE	FUNGSI	BAUD RATE
0	0	0	Register geser	fosc./12
0	1	1	8 bit UART	variable
1	0	2	9 bit UART	fosc./32 atau fosc/64
1	1	3	9 bit UART	variabel

- T1

Sebagai *transmit interrupt flag* yang di-*set* oleh perangkat keras pada akhir dari waktu bit ke 8 pada mode 0, atau pada permulaan dari stop bit pada mode yang lain, dalam setiap pengiriman serial. T1 harus di-*clear* oleh perangkat lunak.

- RI

Sebagai *receive interrupt flag* yang di-*set* oleh perangkat keras pada akhir dari waktu bit ke 8 pada Mode 0, atau separuh dari pemasukan stop bit pada mode yang lain, dalam setiap penerimaan serial. RI harus di-*clear* oleh perangkat lunak.

---

27). Ibid., hal. 8-22.

(MSB) (LSB)

7	6	5	4	3	2	1	0
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

Gambar 2.12.[28]

Register SCON

2.1.13. SERIAL DATA BUFFER REGISTER (SBUF)[29]

Data untuk pengiriman dan dari penerimaan ditempatkan pada serial data buffer register (SBUF). Register SBUF untuk pengiriman adalah berupa register geser 9 bit. Proses menulis data ke SBUF, adalah sama dengan menulis data ke register geser 9 bit dengan byte data menempati 8 bit yang pertama, dengan LSB pada keluaran dari register. Bit ke 9 akan berisi '1' atau sama dengan bit TB8 dari register SCON, tergantung dari mode yang dipilih.

Register-register penerima adalah sebuah register geser dengan panjang 8 bit pada mode 0 atau 9 bit pada mode yang lain, register SBUF, dan sebuah *read only* register yang diisi oleh perangkat keras dengan byte data saat yang bersamaan dengan aktifnya RI. Pada mode UART, bit ke 9 diisikan ke RB8 pada register SCON pada saat yang

---

28). Ibid., hal. 8-22.

29). Ibid., hal. 7-3.

bersamaan dengan pengisian byte data ke dalam SBUF.

2.1.14. PROGRAM STATUS WORD REGISTER (PSW)[3)]

Register program status word (PSW) digunakan untuk menyimpan keadaan (status) dari mikroikontroler dan kontrol operasi untuk mikrokontroler. Pada gambar 2.13 memperlihatkan bit-bit dalam register PSW , di mana bit-bit tersebut nama dan artinya sebagai berikut:

- PSW.7 : CY (*Carry Flag*), diisi '1' atau '0' oleh perangkat keras atau perangkat lunak selama ada instruksi aritmatika.
- PSW.6 : AC (*Auxiliary Flag*), diisi '1' atau '0' selama ada instruksi penambahan atau pengurangan untuk menunjukkan adanya *carry* atau *borrow* dari bit ke-3.
- PSW.5 : F0 (Flag 0), diisi '1' atau '0' atau di-test oleh perangkat lunak sebagai status flag yang didefinisikan oleh pemakai.
- PSW.4 : RS0 (*Register Bank Select Control Bit* 0), diisi '1' atau '0' oleh perangkat lunak untuk menentukan register bank yang bekerja.
- PSW.3 : RS1 (*Register Bank Select Control Bit* 1), diisi '1' atau '0' oleh perangkat lunak untuk menentukan register bank yang bekerja.
- PSW.2 : OV (*Overflow Flag*), diisi '1' atau '0' selama ada instruksi aritmatika oleh perangkat keras untuk

---

3) Ibid., hal. 8-10.

menunjukkan keadaan overflow.

- PSW.1 : cadangan
- PSW.0 : P (*Parity Flag*), diisi '1' atau '0' oleh perangkat keras setiap daur instruksi untuk menunjukkan ganjil/genap dari banyaknya bit '1' pada akumulator.

(MSB) (LSB)

7	6	5	4	3	2	1	0
CY	AC	F0	RS1	RS0	OV	——	P

Gambar 2.13.[31]

Register PSW

2.2. READ ONLY MEMORY (ROM)

Memori jenis ROM bersifat *non volatile*, artinya data yang telah disimpan tidak akan hilang ketika hubungan power supply ke ROM diputuskan. Ada beberapa jenis ROM, tetapi yang paling banyak digunakan sekarang adalah jenis EPROM (*Erasable Programmable Read Only Memory*). EPROM memiliki jalur address yang sesuai dengan kapasitas memorinya. Sebagai contoh, EPROM 2764 memiliki kapasitas 8 kilobyte (= 8192 x 8 = 2^13 x 8) mempunyai 13 jalur address,yaitu: A0 - A12. Pada EPROM terdapat pin $\overline{CS}$ (*Chip Select*) yang berfungsi untuk mengaktifkan rangkaian buffer input/output internal dari EPROM. Pada EPROM ini juga

31). Ibid, hal. 8-10.

terdapat pin $\overline{OE}$ (*output enable*) yang berfungsi untuk mengaktifkan data output dari EPROM.

Pin $\overline{PGM}$ di-*set* '*low*' pada saat EPROM sedang diprogram dan '*high*' pada operasi pembacaan (*read*). Pada pin Vpp diberi tegangan pemrograman yang sesuai ketika waktu pemorograman, misalnya untuk EPROM 2764 adalah 12,5 volt. Saat operasi pembacaan, pin ini umumnya dihubungkan ke +5 volt (untuk EPROM 2732 dihubungkan ke GND/0 volt).

2.3. RANDOM ACCESS MEMORY (RAM)

RAM digunakan untuk menyimpan data sementara waktu. Data pada RAM akan hilang bila supply tegangan terputus. Makanya RAM itu termasuk golongan *volatyle* memory. Berbeda dengan ROM, pada RAM data dapat dibaca atau ditulis. Ada dua jenis RAM, yaitu RAM statis dan RAM dinamis. Pada rangkaian perangkat keras yang direncanakan digunakan RAM statis. Pada pembahasan selanjutnya, cukup disebut RAM saja.

Pada RAM selain terdapat pin $\overline{CS}$ dan $\overline{OE}$ seperti halnya EPROM, juga terdapat $\overline{WE}$ (*write enable*). Pada tabel 2.4 akan ditunjukkan fungsi ketiga pin RAM tersebut secara singkat. Di mana high z maksudnya adalah impedansi tinggi atau dalam keadaan mengambang.

Tabel 2.4.

Fungsi pin $\overline{CS}$, $\overline{OE}$, $\overline{WE}$ pada RAM statis.

$\overline{CS}$	$\overline{OE}$	$\overline{WE}$	MODE	PIN I/O
1	X	X	non aktif	high z
0	0	1	read	out
0	1	0	write	in
0	0	0	write	in

2.4. PROGRAMMABLE PERIPHERAL INTERFACE 8255 (PPI 8255)[32)]

PPI 8255 merupakan *peripheral interface* yang dapat diprogram fungsinya. PPI ini dibagi menjadi dua group, yaitu: group A yang terdiri dari port A (PA0 - PA7) dan port C *upper* (PC4 - PC7); group B yang terdiri dari port B (PB0 - PB7) dan port C *lower* (PC0 - PC3). Pada gambar 2.14 menunjukkan konfigurasi pin IC 8255. Fungsi tiap-tiap pin adalah sebagai berikut:

- D0 - D7:

  Delapan bit data yang merupakan jalur data.

- $\overline{CS}$ (*Chip Select*):

  Saat pin ini ber-*logic low*, maka mikrokontroler akan berhubungan dengan dengan PPI 8255.

- $\overline{RD}$ (Read):

  Saat pin ini ber-*logic low* dan $\overline{CS}$ juga berlogika *low*,

32). Steeman, J. P. M., Data Sheet Book II, PT Elex Media Komputindo, Jakarta, 1988, hal. 239 s. d. 255.

maka PPI 8255 akan mengeluarkan data ke sistem data bus.

- $\overline{WR}$ (*write*):

  Saat pin ini ber-*logic low* dan $\overline{CS}$ juga berlogika *low*, maka sistem bus akan memberikan datanya pada PPI 8255.

- A0 - A1 (Address):

  Address ini menentukan internal register mana dalam 8255 yang akan diterima atau dikirim datanya.

- Reset:

  Logika *high* pada pin reset ini akan mengclearkan kontrol register dan menset semua port (Port A, B, C) ke dalam mode input.

- PA0 - PA7 (Port A):

  Port A digunakan untuk menghubungkan 8255 dengan peralatan luar.

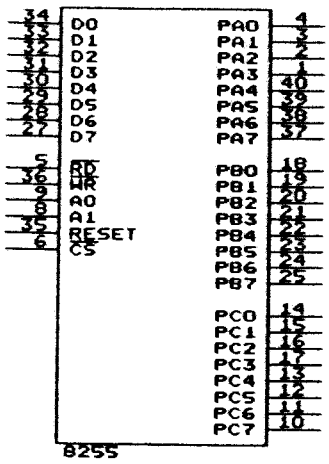
- PB0 - PB7 (Port B):

  Port B mempunyai fungsi sama dengan port A.

- PC0 - PC7 (Port C):

  Port C mempunyai fungsi sama dengan kedua port lainnya. Port C dibagi menjadi dua bagian, yaitu: port C *lower* (PC0 - PC3) dan port C *upper* (PC4 - PC7).

PPI 8255 mempunyai empat buah register, kombinasi address A0 dan A1 menentukan register mana yang bekerja dari keempat register tersebut. Lokasi dan fungsi tiap-tiap register ditunjukkan pada tabel 2.5.



Gambar 2.14.[33]

Konfigurasi pin-pin PPI 8255

Tabel 2.5.[34]

Lokasi dan fungsi register pada PPI 8255

8255A BASIC OPERATION

A_1	A_0	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	INPUT OPERATION (READ)
0	0	0	1	0	PORT A – DATA BUS
0	1	0	1	0	PORT B – DATA BUS
1	0	0	1	0	PORT C – DATA BUS
					OUTPUT OPERATION (WRITE)
0	0	1	0	0	DATA BUS – PORT A
0	1	1	0	0	DATA BUS – PORT B
1	0	1	0	0	DATA BUS – PORT C
1	1	1	0	0	DATA BUS – CONTROL
					DISABLE FUNCTION
X	X	X	X	1	DATA BUS – 3-STATE
1	1	0	1	0	ILLEGAL CONDITION
X	X	1	1	0	DATA BUS – 3-STATE

Control Word Register berfungsi untuk menentukan fungsi dari setiap port dan menentukan mode yang digunakan. 8255 mempunyai tiga mode yang dapat

33). Ibid, hal. 239.

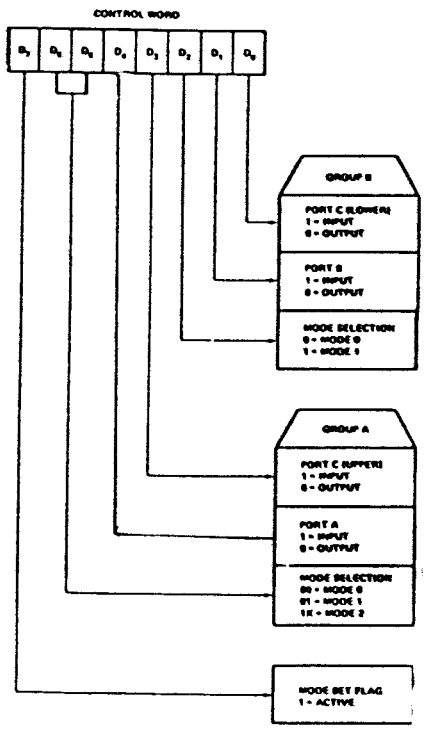
34). Ibid., hal. 240.

dioperasikan, yaitu mode 0 (*basic input/output*), mode 1 (*strobed input/output*), dan mode 2 (*bidirectional bus*). Format dari control word dapat ditunjukkan pada tabel 2.6.

Dalam Tugas Akhir ini PPI 8255 hanya dioperasikan sebagai mode 0. Pada pengoperasian PPI 8255 pada mode 0 tidak diperlukan *handshaking* seperti yang diperlukan pada pengoperasian dengan mode 1 atau mode 2. Pada mode 0 ini port-port PPI 8255 hanyalah berfungsi sebagai input/output saja tergantung format pada *Control Word Register*.

Tabel 2.6.[35)]

Format Control Word Register



35). Ibid, hal. 242.

BAB III
PERENCANAAN

Perencanaan pembuatan tugas akhir ini terdiri dari dua bagian, yaitu perencanaan perangkat keras dan perencanaan perangkat lunak.

Perencanaan perangkat keras akan dibahas antara lain cara kerja rangkaian secara umum, ringkasan rangkaian di sekitar mikrokontroler 8031,dan ringkasan rangkaian output serta rangkaian *display*. Sedangkan untuk perencanaan perangkat lunak akan dibahas antara lain : diagram alir (*flow chart*) dan uraian cara kerja programnya (*software*).

3.1. PERENCANAAN PERANGKAT KERAS

Perangkat keras untuk tugas akhir ini menggunakan IC mikrokontroler 8031 sebagai komponen utamanya. Mikrokontroler ini berfungsi untuk mengontrol proses penerimaan data (dari komputer), proses penyimpanan data (dalam RAM) dan proses pengiriman data melalui PPI ke output lampu-lampu traffic light serta *display seven segment*. Dalam pengoperasiannya, mikrokontroler ini dirangkai dengan komponen-komponen penting lainnya, seperti komponen memory (ROM dan RAM), display lampu-lampu *traffic light* yang melalui PPI 8255, IC - IC TTL yang bekerja sebagai rangkaian logika dan rangkaian dekoder, dan sebagainya.

3.1.1. CARA KERJA RANGKAIAN SECARA UMUM

Komputer IBM PC mengirim data ke mikrokontroler 8031 melalui komunikasi serial *asynchronous*. Data yang dikirim itu berupa input jam, lamanya nyala lampu hijau di setiap perempatan, atau lamanya penyalaan lampu hijau pejalan kaki (pejalan kaki boleh dipakai atau tidak sesuai dengan keinginan). Dalam tugas akhir ini dipakai 3 buah perempatan. Setelah semua data tersebut dikirim maka mikrokontroler tersebut akan memprosesnya. Kemudian data-data tersebut disimpan di RAM luar. Setelah itu dicek input jamnya, apakah jam yang dinputkan itu berupa jam sibuk (kendaraan di jalan raya ramai) atau jam biasa (kendaraan di jalan raya tidak terlalu ramai/banyak) ataukah jam sepi (jumlah kendaraan di jalan raya sedikit).

Lamanya lampu-lampu *traffic light* menyala untuk ketiga kondisi jam tersebut tidak sama, nyalanya lampu-lampu traffic light pada saat jam sibuk tentunya lebih lama dari pada jam biasa. Jam sepi itu terjadi pada jam 11 malam sampai jam 5 pagi, di mana pada jam-jam tersebut kendaraan mulai sedikit/sepi, sehingga lampu kuning saja yang nyala atau mati secara bergantian selama 1 detik. Ketika diketahui keadaan / kondisi jamnya maka mikrokontroler akan mengirim data input yang berupa lamanya nyala lampu-lampu *traffic light* ke output yang berupa lampu-lampu led melalui 8255 PPI. Kemudian setiap satu siklus pada perempatan pertama selesai maka

mikrokontroler akan mengecek kembali kondisi jam, sesudah itu mikrokontroler akan mengirim lagi lamanya nyala lampu-lampu lalu lintas ke output, kejadian ini berlangsung secara terus menerus.

Apabila ada interupsi dari luar yang melalui pin $\overline{\text{INT0}}$ (P1.6), maka mikrokontroler akan berhenti menjalankan instruksi program yang sedang berjalan, untuk melaksanakan interupsi tersebut. Pin $\overline{\text{INT0}}$ (P1.6) dalam tugas akhir ini digunakan sebagai interupsi dari luar yaitu adanya kereta api yang lewat di antara perempatan kedua dan ketiga. Apabila pin $\overline{\text{INT0}}$ diberi *logic* '0' (pin P1.6 berlogika '1'), ini berarti ada kereta api yang lewat maka itu secara hadware lampu merah yang dipasang dekat rel kereta api akan hidup-mati secara bergantian dan secara software lampu-lampu lalu lintas di perempatan kedua dan ketiga yang arah lalu lintasnya menuju ke rel kereta api akan menyala merah sedangkan lampu-lampu lalu lintas lainnya akan menyala normal sesuai dengan program. Setelah tidak ada lagi kereta api yang lewat maka proses penyalaan lampu - lampu lalu lintas akan berjalan normal kembali. Sedangkan pin-pin P1.3, P1.4, P1.5 digunakan untuk memberitahukan ada-tidaknya kecelakaan di perempatan pertama, kedua, dan ketiga. Apabila pin P1.3 berlogika '0', ini berarti ada kecelakaan di perempatan pertama maka itu mikrokontroler akan mengirim data ke output yaitu menyalakan dan mematikan lampu kuning secara bergantian

selama satu detik di jalur yang sedang mendapat lampu hijau sedangkan jalur lainnya di perempatan pertama menyala normal sesuai program. Pada perempatan kedua dan ketiga, penyalaan lampu-lampu lalu lintas berjalan seperti biasanya. Ketika kecelakaan telah dapat diatasi/ditangani, pin P1.3 dikembalikan ke high ,maka proses nyalanya lampu-lampu lalu lintas akan berjalan normal kembali di perempatan pertama. Penyalaan lampu lalu lintas ketika adanya kecelakaan di perempatan kedua (P1.4) dan ketiga (P1.5) analogi dengan kejadian di perempatan pertama.

3.1.2. RANGKAIAN CLOCK

Rangkaian *clock* digunakan untuk mengaktifkan mikrokontroler 8031. Frekuensi *clock* yang dapat diterima oleh 8031 antara 1,2 sampai 12 MHz, baik berupa *clock internal* maupun *eksternal*. Pada peralatan tugas akhir ini digunakan clock *eksternal* yang berupa kristal dengan frekuensi 12 MHz. Dengan frekuensi 12 MHz itu, maka mikrokontroler 8031 memiliki siklus instruksi minimum, yaitu 1 us. Kristal yang digunakan mempunyai frekuensi 12 MHz yang dihubungkan secara pararel anti resonansi.

3.1.3. RANGKAIAN RESET DAN MULTIPLEXER BUS DATA/ALAMAT

Untuk mereset mikrokontroler, pin Reset/Vpd (kaki nomor 9) dijaga pada keadaan tinggi sedikitnya selama 24 periode osilator. Pada rangkaian ini frekuensi osilatornya

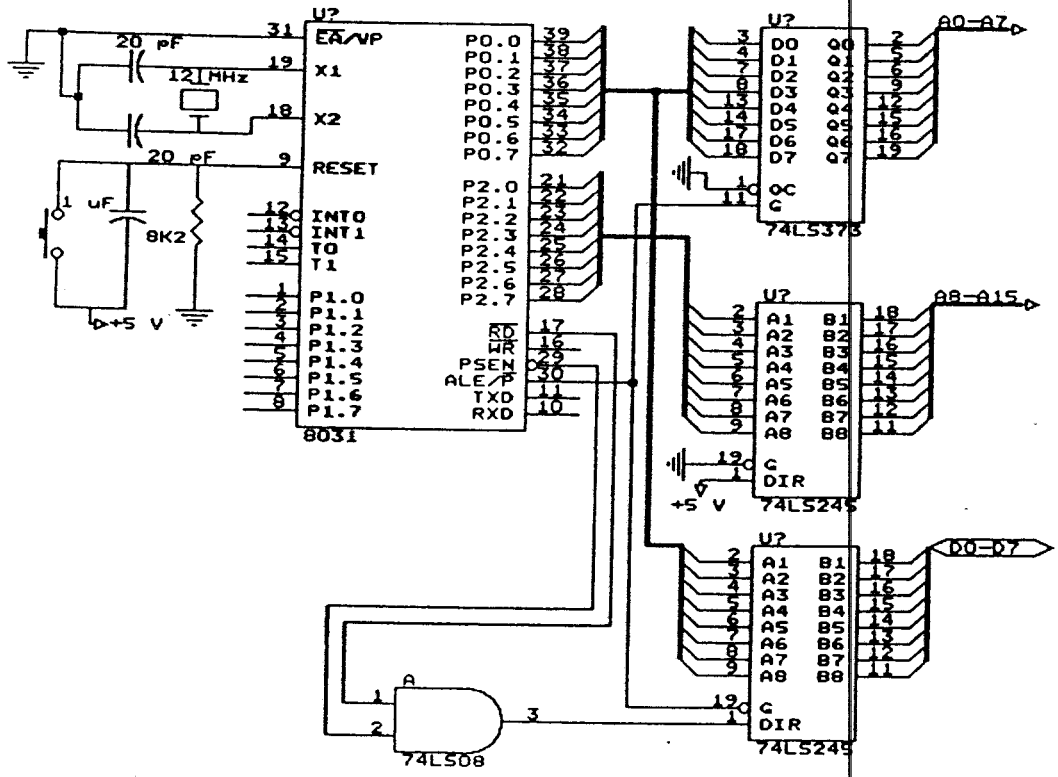
adalah 12 MHz, sehingga periodanya adalah sekitar 83 ns. Jadi untuk mereset mikrokontroler, pin Reset/Vpd harus dijaga pada keadaan tinggi paling sedikit selama

= 24 x T osilator

= 24 x 83 ns

= 1992 ns

Untuk memenuhi kriteria itu dibuat rangkaian reset dengan sebuah saklar yang dihubungkan dengan sumber tegangan 5 V, sebuah tahanan dan sebuah kapasitor, seperti yang ditunjukkan pada gambar 3.1.



Gambar 3.1.

Rangkaian clock, reset, dan multiplex data/alamat

Pada saat saklar ditekan, arus mengalir melaluitahanan dan mengisi kapasitor, sehingga tegangan pada pin Reset naik menjadi mendekati tegangan sumber. Setelah reset dilakukan, mikrokontroler menghentikan eksekusi instruksi dan tidak aktif. Semua register direset ke keadaan awalnya, dan operasi normal dilakukan lagi mulai dari lokasi program memori 0000H.

Rangkaian untuk me-*multipleks* antara bus data dengan bus alamat rendah ditunjukkan pada gambar 3.1. Rangkaian ini diperlukan karena bus data dan bus alamat rendah dari 8031 menjadi satu pada port 0. Untuk keperluan ini digunakan IC 74ls373 dan IC 74ls245.

IC 74ls373 fungsinya untuk menampung 8 bit alamat rendah (A0 - A7). Kaki kontrol outputnya ($\overline{OC}$, pin nomor 1) selalu berada pada keadaan rendah (dihubungkan ke ground), sedangkan Kaki *enable*-nya (G, pin nomor 11) dihubungkan ke kaki ALE 8031(pin nomor 30). Pada saat 8031 mengakses memori *eksternal*, ALE aktif, sehingga mengaktifkan *enable* 74ls373 dan me-*latch* bus alamat rendah ke sistem bus data rangkaian. IC 74ls245 fungsinya menampung 8 bit data (D0 - D7). Kaki *enable*-nya ($\overline{G}$, pin nomor 11) dihubungkan ke kaki ALE 8031, bila ALE aktif maka 74ls245 tidak berfungsi ketika ALE ber-*logic low* maka 74ls245 bekerja untuk melewatkan data bus yang mana arah *transfer* data busnya tergantung dari kaki DIR (pin nomor 1). Apabila DIR berlogic high maka data dari mikrokontroler di-transfer ke

device luar sedangkan bila DIR berlogika rendah maka data dari *device* luar di-*transfer* ke mikrokontroler. *Device* dalam hal ini berupa RAM, ROM, PPI 8255. Logika kaki DIR ditentukan oleh kaki RD (pin nomor 17) dan PSEN (pin nomor 29) dari mikrokontroler 8031.

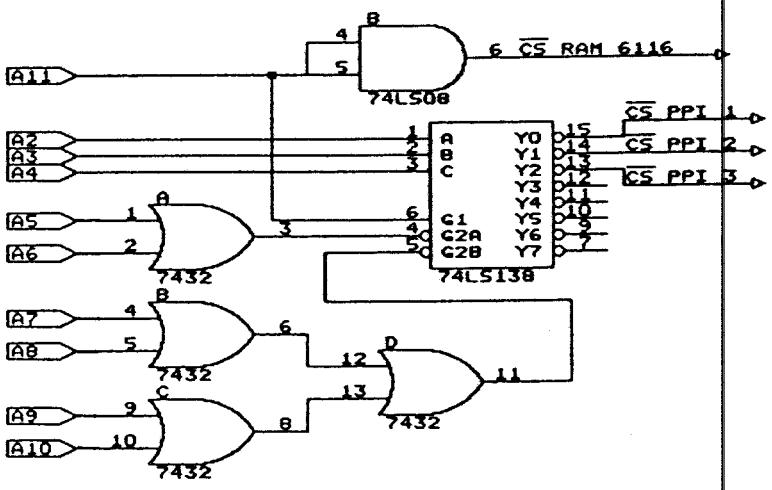
Port 2 ini berfungsi sebagai bus alamat tinggi oleh karena itu port 2 dibuffer dengan 74ls245. Pembufferan ini diperlukan untuk menghindari kelebihan beban pada port 2 ini, karena port ini dihubungkan dengan lebih dari satu beban setara TTL (port 2 maksimum men-*drive* 1 beban setara TTL). Pin *enable*-nya (G, kaki nomor 19) dihubungkan ke *ground*, dan pin kontrol arahnya (DIR, kaki nomor 1) dihubungkan ke sumber tegangan 5 volt.

3.1.4. RANGKAIAN DECODER

Rangkaian decoder ini dimaksudkan agar hanya satu device saja yang bekerja. Device yang dibicarakan di sini berupa RAM dan PPI 8255. Dalam Tugas Akhir ini PPI 8255 diberlakukan sebagai RAM luar oleh karena itu supaya tidak terjadi overlapping RAM 6116 maka PPI 8225 mempunyai alamat yang berlainan dengan RAM 6116.

Di sini digunakan 3 buah PPI 8255 untuk menampilkan output yang berupa lampu led dan seven segment. Alamat RAM 6116 ditentukan adalah dari 000H - 7FFH (2 kilobyte), sedangkan PPI 1 beralamat mulai dari 800H - 803H ; PPI 2 beralamat mulai dari 804H - 807H; PPI 3 beralamat mulai

dari 808H - 80BH. Dari alamat-alamat tersebut di atas terlihat bahwa addres A11 yang dapat membedakan RAM ataukah PPI 8255 yang bekerja. Bila A11 berlogika '0' maka RAM 6116 yang bekerja sedangkan bila A11 berlogika '1' maka PPI yang bekerja. Untuk lebih jelasnya dapat dilihat pada gambar 3.2. yang menunjukkan rangkaian decoder.



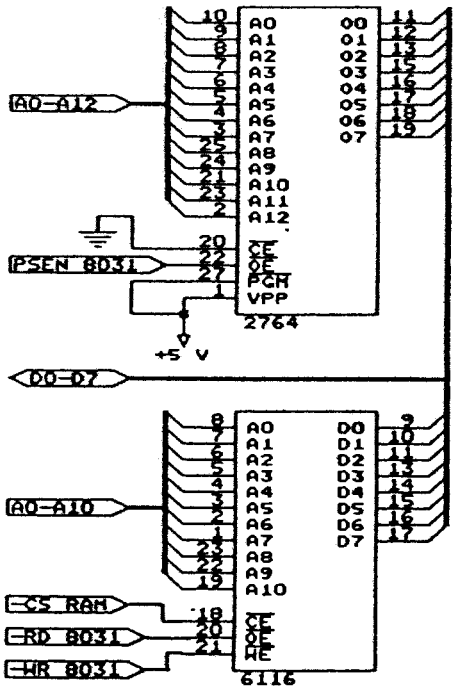
Gambar 3.2.

Rangkaiaan decoder

3.1.5. RANGKAIAN MEMORI PROGRAM

Rangkaian memori program berfungsi untuk menyimpan program yang menjalankan sistem rangkaian. Memori program ini menggunakan EPROM 2764 yang mempunyai kapasitas memori 8 kilo byte, alamatnya mulai dari 0000H - 1FFFH. Rangkaian memori program ditunjukkan pada gambar 3.2. Input bus alamatnya dihubungkan dengan bus 8031, demikian juga bus datanya. Kaki *chip enable* ($\overline{CE}$) ROM dihubungkan ke ground,

kaki output enable-nya dihubungkan ke sinyal $\overline{\text{PSEN}}$ 8031, sedangkan kaki $\overline{\text{PGM}}$ dan Vpp dihubungkan ke tegangan 5 Volt.



Gambar 3.3.

Rangkaian memori program dan data eksternal

Cara kerja dari rangkaian memori program *eksternal* ini adalah sebagai berikut: pada saat mengakses lokasi tertentu dari memori program, mikrokontroler mengirimkan alamat dari lokasi memori tersebut. Setelah itu mikrokontroler mengeluarkan sinyal $\overline{\text{PSEN}}$ (*Program Store Enable*) yang kemudian meng-*enable* EPROM 2764 sehingga isi dari alamat memori program yang dimaksud itu telah berada pada data bus mikrokontroler 8031 dan siap untuk dijalankan/ dilaksanakan.

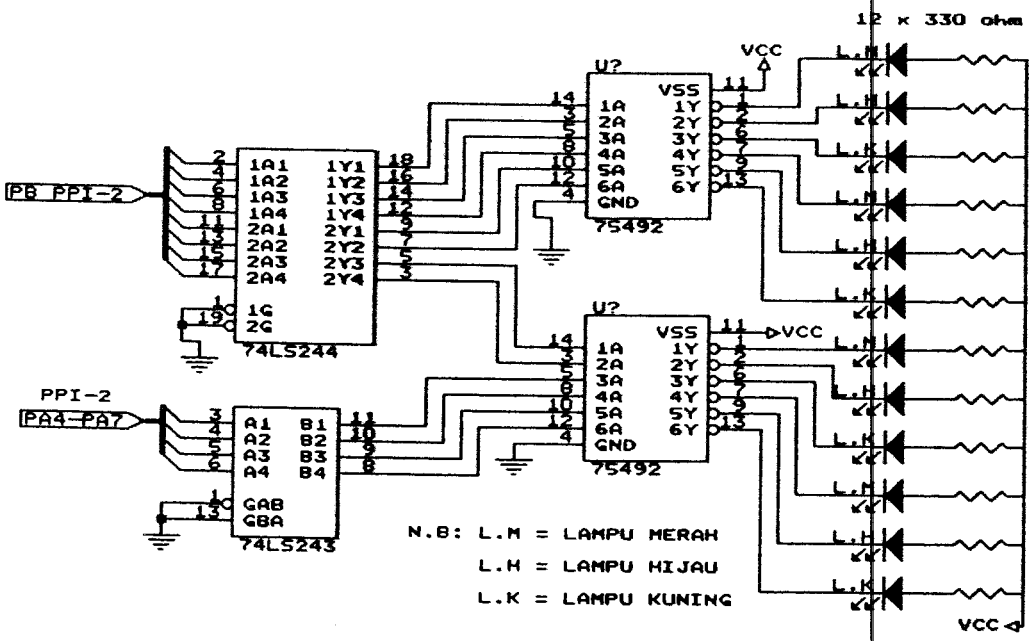
3.1.6. RANGKAIAN MEMORI DATA EKSTERNAL

Rangkaian memori data berfungsi untuk menyimpan data yang masuk. Memori data eksternal menggunakan RAM statis 6116, dan RAM ini mempunyai kapasitas sebesar 2 kilobyte, yang alamatnya mulai dari 000H - 7FFH. Input bus alamatnya dihubungkan dengan bus alamat 8031 melalui buffer IC 74LS373 dan IC 74ls245 , dan 8 bus datanya juga dihubungkan dengan 8031 melalui buffer IC 74LS245. Demikian pula dengan input $\overline{OE}$ dan $\overline{WE}$ dari RAM 6116 secara berturut-turut dihubungkan ke kaki $\overline{RD}$ dan $\overline{WR}$ mikrokontroler 8031. Sedangkan pin 18 ,yaitu $\overline{CE}$ berfungsi untuk mengaktifkan dan menonaktifkan RAM. $\overline{CE}$ diberi *logic* '0' maka RAM akan aktif dan bila diberi *logic* '1' maka RAM tidak bekerja (non aktif). Untuk mengaktifkan $\overline{CE}$ digunakan decoding seperti yang telah dijelaskan dalam rangkaian decoder. Dilakukan decoding ini disebabkan karena dalam tugas akhir digunakan juga PPI 8255 yang diberlakukan sebagai RAM luar. Rangkaian memori data eksternal ditunjukkan dalam gambar 3.3.

3.1.7. RANGKAIAN DISPLAY

PPI yang mana dalam perencanaan alat ini difungsikan sebagai RAM luar sehingga pin A0, A1, $\overline{RESET}$, $\overline{RD}$, $\overline{WR}$ dihubungkan dengan mikrokontroler 8031. Pin $\overline{CS}$ dari 8031 berfungsi sama seperti pin $\overline{CE}$ dari RAM. Jadi apabila $\overline{CE}$ mendapat *logic* '0' maka PPI berfungsi atau aktif dan bila mendapat *logic* '1' maka PPI tidak aktif atau tidak

berfungsi. Untuk mengaktifkan pin $\overline{CE}$ RAM dan $\overline{CS}$ PPI digunakan sistem decoding. PPI yang dipakai sebanyak 3 buah dan alamatnya dari 800H sampai 80BH. PPI 1 alamatnya dari 800H sampai 803H, PPI 2 alamatnya dari 804H sampai 807H, PPI 3 alamatnya dari 808H sampai 80BH.



Gambar 3.4.

Rangkaian output berupa lampu-lampu lalu lintas

Ketiga buah PPI ini diset pada mode 0 yang mana Port A, Port B, dan Port C (*upper* dan *lower*) digunakan sebagai output/*display* yang berupa lampu-lampu led untuk menampilkan lampu-lampu *traffic light* dan seven segment untuk menampilkan jam dalam sehari. Sesuai dengan hal di atas maka *control word* untuk PPI 8255 adalah 10000000B (80H). Uraian tentang format *control word* dapat dilihat

pada tabel 2.6.

Sebelum port output dari PPI 8255 digunakan untuk mengemudikan lampu led, diperlukan dua tahap buffering. Buffering tahap pertama dilakukan oleh IC *unidirectional* buffer 74LS244. Buffer kedua adalah IC 75492. Blok diagram rangkaian output lampu-lampu led untuk perempatan pertama dapat dilihat pada gambar 3.4. Untuk perempatan kedua dan ketiga rangkaiannya analogi dengan rangkaian di perempatan pertama, untuk rangkaian output ke lampu-lampu led ketiga buah perempatan dapat dilihat di lampiran.

Sedangkan untuk seven segment, keluaran port output dari PPI 8255 diteruskan ke input IC 74ls48 (IC pengubah bilangan biner ke bilangan desimal) kemudian dari output IC 74ls48 ditampilkan ke seven segment melalui sebuah resistor. Di mana harga resistor tersebut dapat dicari sebagai berikut:

Rmin = (Vcc - 0,7) / Iol

Di mana Iol dari IC 74ls48 = 8mA

sehingga Rmin = 237,5 ohm

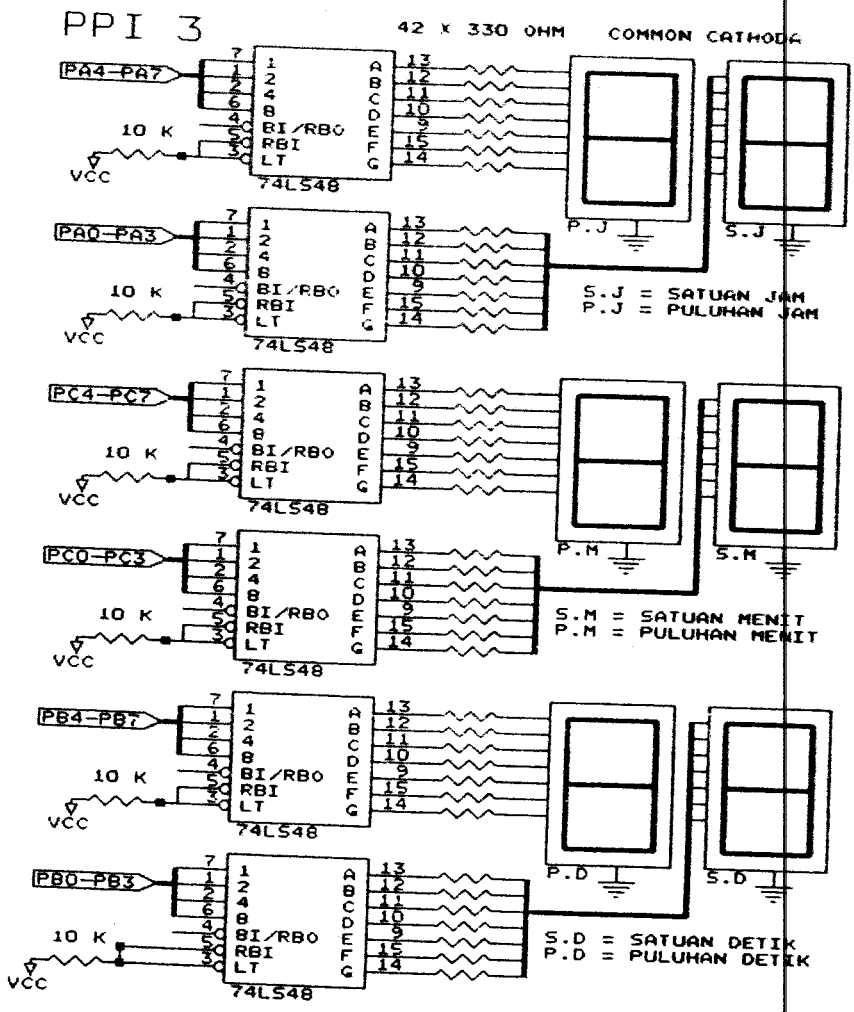
Rmax = (Vcc - 0,7) / Iih

Di mana Iih dari IC 74ls48 = 400 uA

sehingga Rmax = 4750 ohm

Di mana 0,7 volt adalah tegangan led normal yang berada dalam seven segment. Jadi 237,5 ohm < R < 4750 ohm , maka itu pemilihan nilai R sebesar 330 ohm cukup memadai. Rangkaian *display seven segment* ditunjukkan pada gambar

3.5.



Gambar 3.5.

Rangkaian *Display* jam dalam sehari

3.1.8. PEMBUATAN RUMUS WAKTU SINKRONISASI

Untuk mencapai waktu sinkronisasi pada jalur prioritas tiga buah perempatan maka harus lamanya siklus pada perempatan pertama sama dengan lamanya siklus pada perempatan kedua sama dengan lamanya siklus pada perempatan

ketiga. Secara matematis dapat ditulis sebagai berikut:

$$1A + 1B + 1C = 2A + 2B + 2C = 3A + 3B + 3C$$

Di mana angka 1, 2, dan 3 dari rumus di atas menyatakan perempatan pertama, kedua dan ketiga. Sedangkan A, B, dan C menyatakan posisi di tiap-tiap perempatan. 1A, 2A, dan 3A adalah lamanya nyala lampu hijau di perempatan pertama, kedua dan ketiga pada posisi A ; 1B, 2B, dan 3B adalah lamanya nyala lampu hijau di perempatan pertama, kedua dan ketiga pada posisi B ; 1C, 2C, dan 3C adalah lamanya nyala lampu hijau di perempatan pertama, kedua dan ketiga pada posisi C. Dari gambar maket (dapat dilihat pada lampiran) yang direncanakan, 1A, 2A, 3A adalah jalur prioritas maka dapat dibuat rumus sebagai berikut:

$$2A = 1A + 1B + L.K$$

$$1A + 1B + 1C = 2A + 2B + 2C$$

$$1A + 1B + 1C = 1A + 1B + L.K + 2B + 2C$$

$$1C = 2B + 2C + L.K$$

$$3A = 2A + 2B + L.K$$

$$2A + 2B + 2C = 3A + 3B + 3C$$

$$2A + 2B + 2C = 2A + 2B + L.K + 3B + 3C$$

$$2C = 3B + 3C + L.K$$

Diambil $3B = 3C$

Di mana L.K adalah lampu kuning. Dari hasil *survey* ternyata nyala lamanya lampu kuning tetap (*konstan*), yaitu

selama 3 detik. Kemudian 1A, 1B, 2B, 3B berupa variabel (pengisian data dari komputer disesuaikan dengan jumlah kendaraan yang ada pada jam biasa atau jam sibuk). Di sini juga dipakai all red selama 1 detik setiap pergantian posisi penyalaan lampu hijau pada semua perempatan, tujuannya untuk menghindari kecelakaan selama pergantian posisi penyalaan lampu lalu lintas. Hasil perencanaan dan pengisian lama nyalanya lampu-lampu hijau di setiap perempatan berdasarkan rumus di atas adalah sebagai berikut:

Pada jam biasa:

1A = 12 detik

1B = 9 detik

2B = 9 detik

3B = 9 detik

Dari pengisian diatas didapatkan:

1A = 12 detik	2A = 24 detik	3A = 34 detik
1B = 9 detik	2B = 9 detik	3B = 9 detik
1C = 33 detik	2C = 21 detik	3C = 9 detik

Pada jam sibuk:

1A = 15 detik

1B = 11 detik

2B = 11 detik

3B = 11 detik

Dari pengisian diatas didapatkan:

1A = 15 detik	2A = 29 detik	3A = 43 detik
1B = 11 detik	2B = 11 detik	3B = 11 detik
1C = 39 detik	2C = 27 detik	3C = 11 detik

Dalam tugas akhir ini dibuat jarak antara satu perempatan dengan perempatan lainnya maksimum 9,9 km, jadi jarak berapapun boleh asal di bawah 9,9 km. Begitupun dengan kecepatan kendaraan (maksimum 65 km/jam). Tapi sebaiknya mengikuti peraturan atau kenyataan di jalan raya. Kecepatan kendaraan di jalan raya menurut peraturan lalu lintas adalah 40 km/jam. Untuk mencapai kecepatan stabil 40 km/jam diperlukan percepatan yang bergerak dari 0 sampai 40 km/jam. Waktu yang ditempuh dari Vo = 0 menuju ke kecepatan stabil Vt = 40 km/jam adalah 8,8 detik (hasil ini diambil dari hasil percobaan sebanyak 10 X, kemudian diambil rata-ratanya). Kemudian diterapkan rumus sebagai berikut:

$$a = \frac{Vt - Vo}{tp} = \frac{Vt}{tp} = \frac{40 \text{ km/jam}}{8,8 \text{ detik}}$$

$$Sp = Vo.t + (1/2).a.tp^2$$

$$Sp = (1/2).a.tp^2 = (1/2) \text{ . } 40 \text{ Km/Jam . } (8,8 \text{ detik})$$

$$= 48,89 \text{ m}$$

Seumpama S1 (jarak perempatan pertama dan kedua) diinputkan lebih besar dari Sp maka rumus yang dipakai adalah sebagai berikut:

$$T1 = tp + \frac{S1 - Sp}{Vt}$$

$$T2 = S2 / Vt$$

Contoh: S1 = 100 m ; S2 = 200 m ; Vt = 40 Km/jam

$$\text{Jadi } T1 = 8{,}8 \text{ detik} + \frac{(100 - 48{,}89)\ m}{40\ Km/jam} = 13 \text{ detik}$$

$$T2 = S2/Vt = 200\ m / (40\ km/jam) = 18 \text{ detik}$$

Di mana S2 adalah jarak antara perempatan kedua dan ketiga, T1 adalah waktu yang ditempuh dari perempatan pertama sampai ke perempatan kedua, T2 adalah waktu yang ditempuh dari perempatan kedua sampai ke perempatan ketiga, sedangkan Vt adalah kecepatan stabil yang diinginkan (berupa input).

Seumpama S1 < Sp, maka dapat diterapkan rumus sebagai berikut:

$$S1 = (1/2).a.T1^2$$

$$T1^2 = \frac{2.S1}{a}$$

$$T1 = \sqrt{(2.S1)/a}$$

*). Bila: $S1 + S2 > Sp$

$$Tx = tp + \frac{S1 + S2 - Sp}{Vt}$$

$$T2 = Tx - T1$$

*). Bila: S1 + S2 < Sp

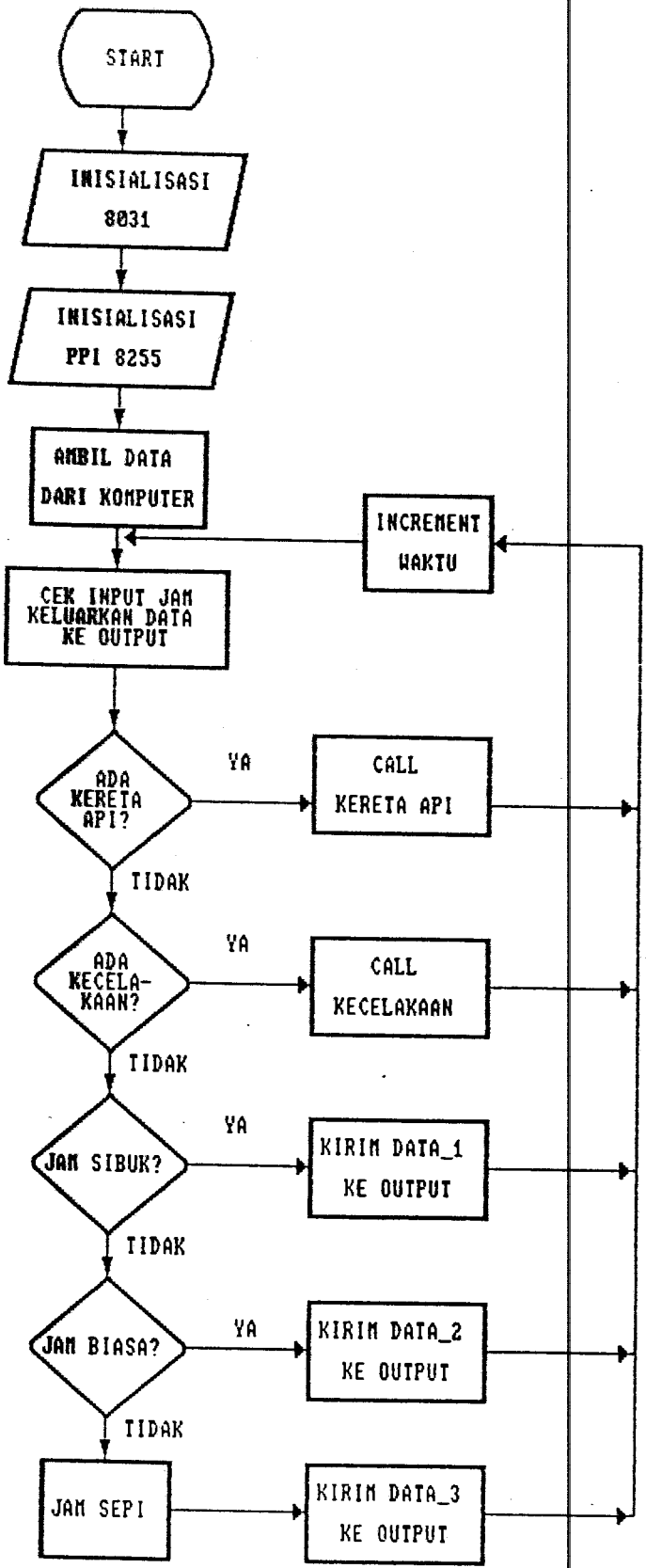
$$Tx = \frac{2.(S2+S1)}{a}$$

$$T2 = Tx - T1$$

3.2. PERENCANAAN PERANGKAT LUNAK

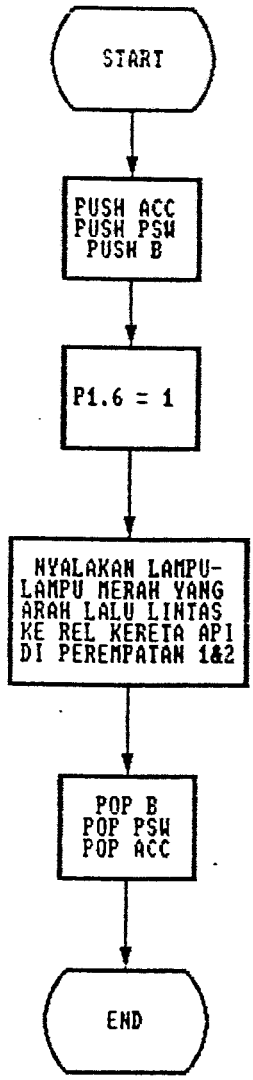
Pada perencanaan perangkat lunak dipakai 2 bahasa program, yaitu bahasa Turbo Pascal dan bahasa Assembly mikrokontroler 8031. Bahasa Pascal dibuat hanya untuk men-*transfer* data (yang berupa: input jam, lamanya nyala lampu-lampu *traffic light*) dari komputer ke mikrokontroler. Sedangkan proses selanjutnya dikerjakan oleh mikrokontroler 8031 dengan menggunakan perangkat lunak mikrokontroler itu sendiri, kemudian dengan bantuan perangkat lunak MCS-51 *cross assembler* diubah ke dalam bahasa mesin dan dimasukkan ke dalam EPROM. *Flowchart* program utama, prosedur interrupt kereta api, prosedur kecelakaan lalu lintas ditunjukkan berturut-turut pada gambar 3.6, gambar 3.7, gambar 3.8. Dalam gambar 3.6 menyatakan data 1, data 2, data 3 adalah lamanya nyala lampu-lampu lalu lintas di setiap perempatan. Di mana data 1 merupakan lama nyalanya lampu-lampu lalu lintas pada jam sibuk dan data 2 merupakan lama nyalanya lampu-lampu lalu lintas pada jam biasa sedangkan data 3 merupakan lama nyalanya pada jam sepi. Tentunya lamanya lampu-lampu lalu lintas menyala pada jam sibuk lebih lama

dari pada jam biasa. Dan pada jam sepi lampu kuning menyala kedap-kedip di setiap perempatan. Sedangkan pada gambar 3.8 ada tertulis lampu kuning kedap-kedip pada jalur on di perempatan pertama atau kedua atau ketiga. Maksud dari kalimat tersebut yaitu bahwa apabila ada kecelakaan yang terjadi di perempatan pertama atau kedua atau ketiga di mana pada saat itu suatu jalur dalam keadaan lampu hijau, maka pada jalur tersebut akan terjadi lampu kuning kedap-kedip selama kecelakaan masih terjadi sesudah kecelakaan dapat teratasi/diselesaikan maka lampu-lampu lalu lintas akan berjalan normal kembali.



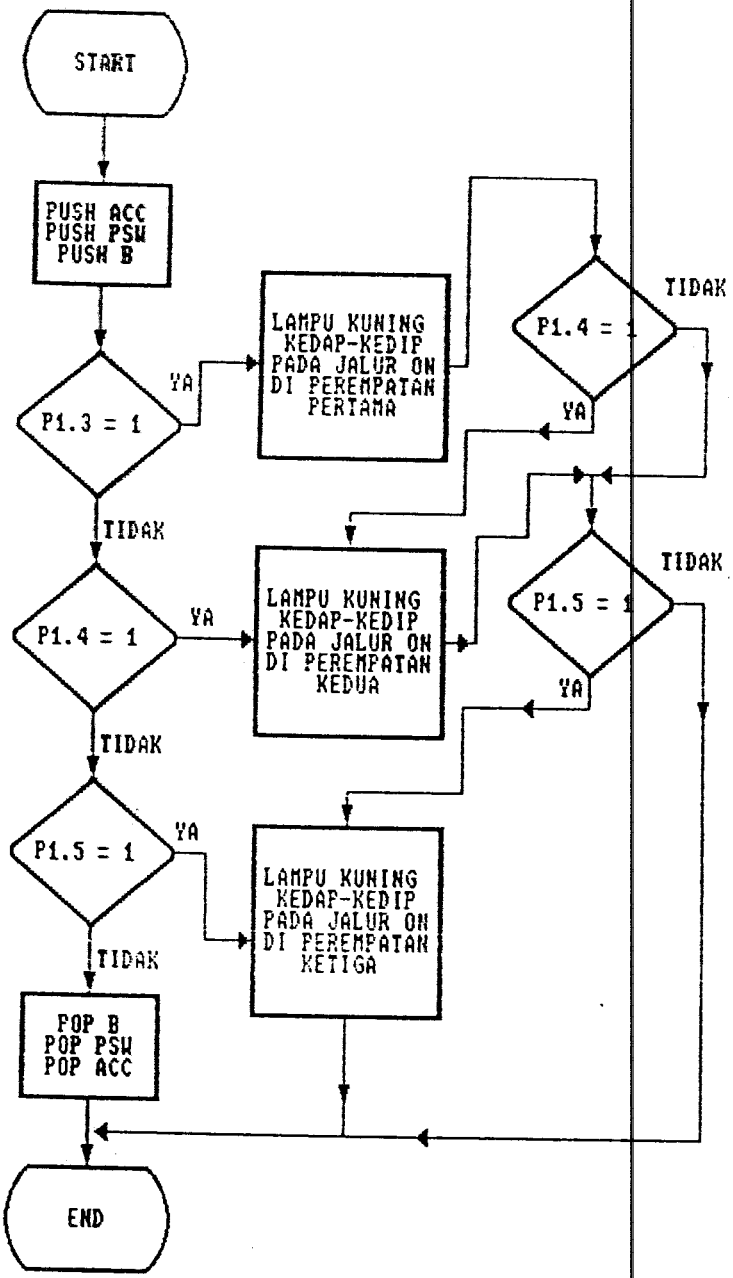
Gambar 3.6.

Flowchart Program Utama



Gambar 3.7.

Flowchart adanya kereta api



Gambar 3.8.

Flowchart kecelakaan lalu lintas

3.2.1. RINGKASAN PROGRAM UTAMA

Proses inisialisasi mikrokontroler 8031 dan persiapan kondisi awal dari peralatan Tugas Akhir dalam menerima data dari komputer dilaksanakan dalam program utama. Setelah reset dilakukan maka isi dari masing-masing register dalam mikrokontroler 8031 sebagai berikut: register SP berisi 07H, register IE berisi 60H, register IP berisi E0H, register-register lainnya berisi 00H, dan port 0 sampai port 3 berisi FFH artinya adalah semua port difungsikan sebagai input (masukan). Untuk mengubah semua nilai tersebut supaya sesuai dengan keinginan program maka itu dilakukan inisialisasi yang lokasinya mulai dari 0000H sampai 0003H pada mikrokontroler 8031.

Inisialisasi yang dilakukan oleh 8031 adalah dengan mengubah nilai register SP, IP, dan IE. Dalam inisialisasi ini SP diisi dengan nilai 50H sehingga register R0 sampai R7 dapat digunakan di bawah lokasi 50H. INT0 (adanya kereta api) mempunyai prioritas yang lebih tinggi oleh karena itu register IE harus berisi 9BH dan register IP harus berisi 01H. Setelah inisialisasi dilakukan maka perlu persiapan dari peralatan untuk menerima data dari komputer selanjutnya mikrokontroler 8031 akan memprosesnya kemudian hasil proses tersebut dikeluarkan ke output.

BAB IV
PENGUKURAN ALAT

4.1. PENGUJIAN DAN PENGUKURAN ALAT

Setelah melakukan pembuatan alat tugas akhir ini tentu harus mengukur dan menguji kemampuan alat ini. Pengujian yang dilakukan adalah pengujian kemampuan peralatan dalam menerima data dari komputer IBM melalui komunikasi serial *asynchronous*. Sedangkan pengukuran yang dilakukan adalah pengukuran lamanya penyalaan setiap lampu merah, kuning, hijau pada semua perempatan.

4.2. PENGUKURAN LAMA PENYALAAN LAMPU LAMPU LALU LINTAS

Lamanya penyalaan lampu-lampu lalu lintas diukur dengan *stop watch*. Diadakan pengukuran ini maksudnya adalah untuk membandingkan antara hasil perencanaan dan pengukuran apakah sesuai atau tidak. Perbandingan antara perencanaan dan hasil pengukuran lama penyalaan lampu-lampu lalu lintas pada jam sepi dan lama penyalaan lampu-lampu pejalan kaki untuk semua perempatan dapat dilihat pada tabel 4.1. Sedangkan lama penyalaan lampu-lampu merah, hijau, dan kuning untuk perempatan pertama, kedua, ketiga secara berturut-turut dapat dilihat pada tabel 4.2, tabel 4.3, tabel 4.4. Di mana hasil perencanaan ini diambil dari salah satu pengisian data dari komputer pada jam biasa, jam sibuk serta jam sepi.

Tabel 4.1

Lama penyalaan lampu-lampu lalu lintas

pada jam sepi dan

lama penyalaan lampu-lampu pejalan kaki

pada semua perempatan

Kondisi	Posisi	Nyala lampu	Hasil	
			Perencanaan (detik)	Pengukuran (detik)
Jam Sepi Perempat-an 1,2,3	A/B/ C/D	Kuning	1	1
		Kuning Padam	1	1
Lampu Pejalan Kaki Perempatan 1		Merah	67	67
		Hijau	6	6
Lampu Pejalan Kaki Perempatan 2		Merah	67	67.01
		Hijau	6	6
Lampu Pejalan Kaki Perempatan 3		Merah	67	67.02
		Hijau	6	6

Tabel 4.2.

Lama penyalaan lampu-lampu lalu lintas

di perempatan pertama

Kondisi	Posisi	Nyala lampu	Hasil	
			Perencanaan (detik)	Pengukuran (detik)
Jam Biasa	A	Merah	58	58.02
		Hijau	12	12
		Kuning	3	3.01
	B	Merah	61	61
		Hijau	9	9.01
		Kuning	3	3.01
	C	Merah	37	37
		Hijau	33	33.02
		Kuning	3	3
Jam Sibuk	A	Merah	66	66
		Hijau	15	15.01
		Kuning	3	3
	B	Merah	70	70.02
		Hijau	11	11.01
		Kuning	3	3.01
	C	Merah	42	42.01
		Hijau	39	39.01
		Kuning	3	3

Tabel 4.3.

Lama penyalaan lampu-lampu lalu lintas

di perempatan kedua

Kondisi	Posisi	Nyala lampu	Hasil	
			Perencanaan (detik)	Pengukuran (detik)
Jam Biasa	A	Merah	46	46.02
		Hijau	24	24.01
		Kuning	3	3.01
	B	Merah	61	61.02
		Hijau	9	9.01
		Kuning	3	3.01
	C	Merah	49	49.01
		Hijau	21	21.02
		Kuning	3	3
Jam Sibuk	A	Merah	52	52.01
		Hijau	29	29.01
		Kuning	3	3
	B	Merah	70	70.01
		Hijau	11	20.03
		Kuning	3	3.01
	C	Merah	54	54.01
		Hijau	27	27.02
		Kuning	3	3

Tabel 4.4.

Lama penyalaan lampu-lampu lalu lintas

di perempatan ketiga

Kondisi	Posisi	Nyala lampu	Hasil	
			Perencanaan (detik)	Pengukuran (detik)
Jam Biasa	A	Merah	34	34
		Hijau	36	36.01
		Kuning	3	3
	B	Merah	61	61.01
		Hijau	9	9
		Kuning	3	3.01
	C	Merah	61	61.02
		Hijau	9	9.01
		Kuning	3	3
Jam Sibuk	A	Merah	38	38.01
		Hijau	43	43.02
		Kuning	3	3.01
	B	Merah	70	70.02
		Hijau	11	11.01
		Kuning	3	3
	C	Merah	70	70.03
		Hijau	11	11.01
		Kuning	3	3

BAB IV
PENUTUP

4.1. KESIMPULAN

Dari hasil perencanaan dan pengukuran alat dapat diambil kesimpulan sebagai berikut:

- Peralatan tugas akhir yang berjudul 'Perencanaan dan pembuatan prototype sinkronisasi *traffic light* dengan menggunakan mikrokomputer 8031' dapat digunakan / ditempatkan di mana saja dan bersifat *programable* artinya waktu sinkronisasi dapat diprogram sesuai dengan keinginan/kebutuhan.
- Dengan peralatan sinkronisasi *traffic control* ini diharapkan dapat membuat sinkron atau selaras beberapa buah *traffic light* yang berada pada satu jalur (jalur prioritas) dengan jarak berapapun, sehingga dapat mengurangi kemacetan lalu lintas pada jalur prioritas tersebut.
- Penggantian program bisa dilaksanakan dalam waktu yang singkat, dengan hanya mengganti 1 IC yang berisi program, yaitu EPROM. Hal ini sangat membantu agar kelancaran lalu lintas tetap teratur selama penggantian program tersebut.
- Fasilitas-fasilitas yang tersedia di dalam mikrokontroler 8031 ini sangat menguntungkan karena dapat mengontrol peralatan tugas akhir ini. Fasilitas-fasilitas itu tersebut adalah pararel dan

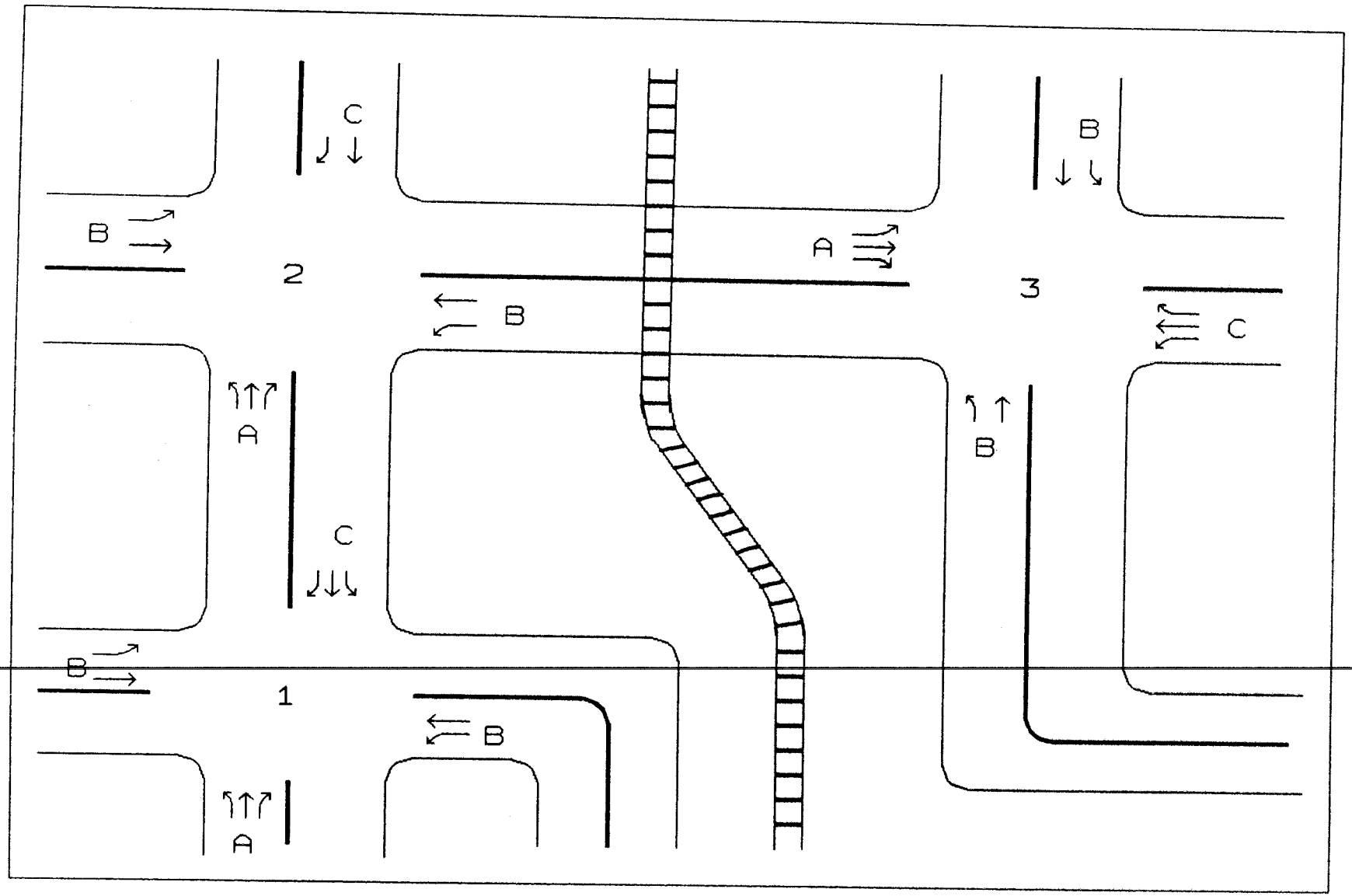
serial I/O, RAM dalam, memori program dan memori data luar (masing-masing mempunyai kemapuan sampai 64 kilobyte), dua buah *timer/counter* 16 bit, dan sebagainya.

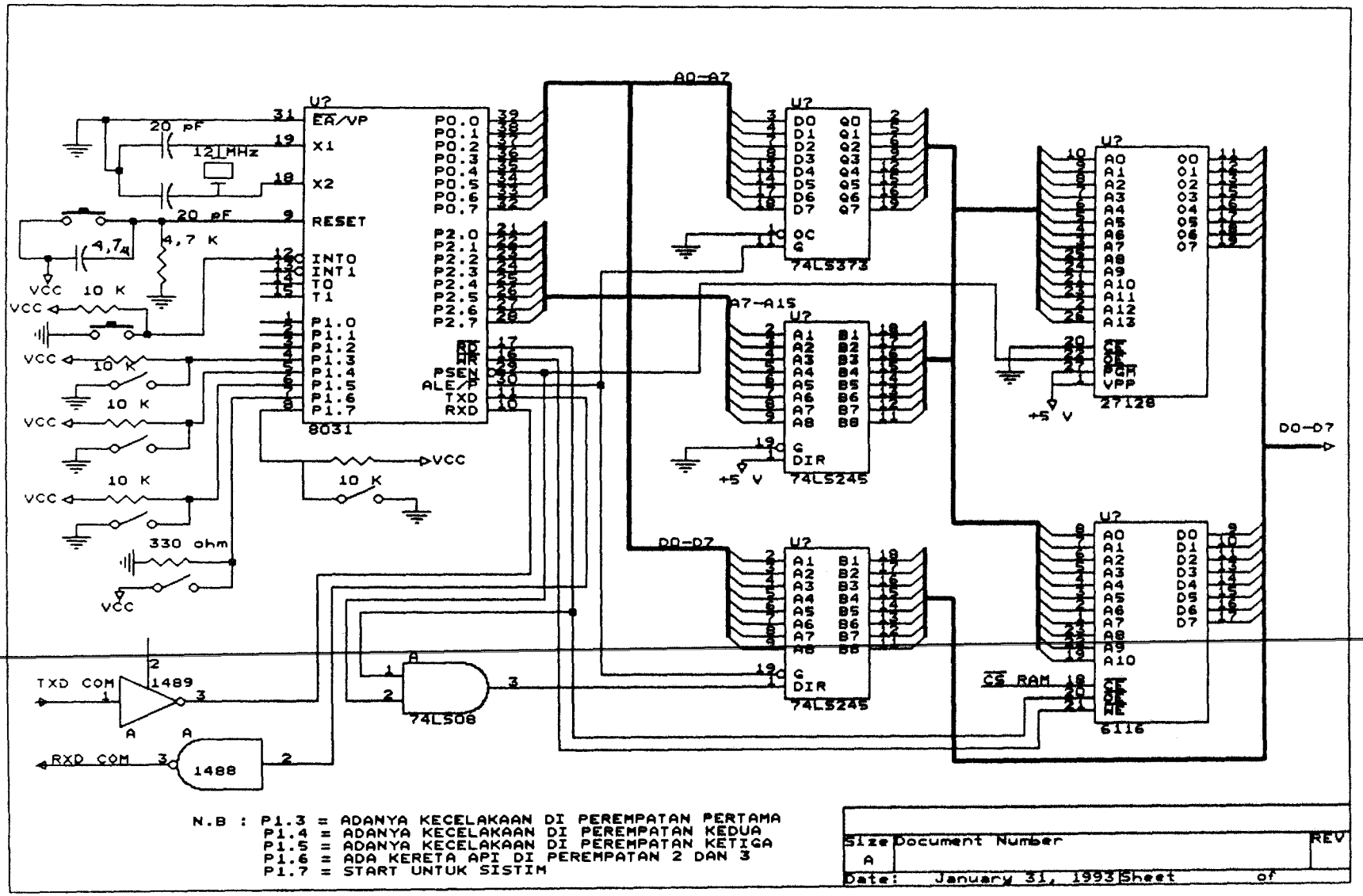
5.2. SARAN

Akhir kata Penulis tidak luput dari kekurangan-kekurangan sehingga diharapkan agar untuk meningkatkan kemampuan sinkronisasi *traffic light* ini dapat dilakukan dengan penyempurnaan *software* untuk mengoptimumkan unjuk kerja peralatan tersebut. Selain itu dalam kenyataan di jalan raya kota Surabaya untuk jalur prioritas, jarak maksimum antara satu perempatan dengan perempatan lainnya adalah 1,8 km. Untuk jarak yang lebih dari 1,8 km kemungkinan waktu sinkronisasi tidak tercapai karena adanya kendaraan dari arah-arah yang lain yang menuju ke jalur prioritas. Semoga Tugas Akhir ini ada manfaatnya bagi kita semua.

DAFTAR PUSTAKA

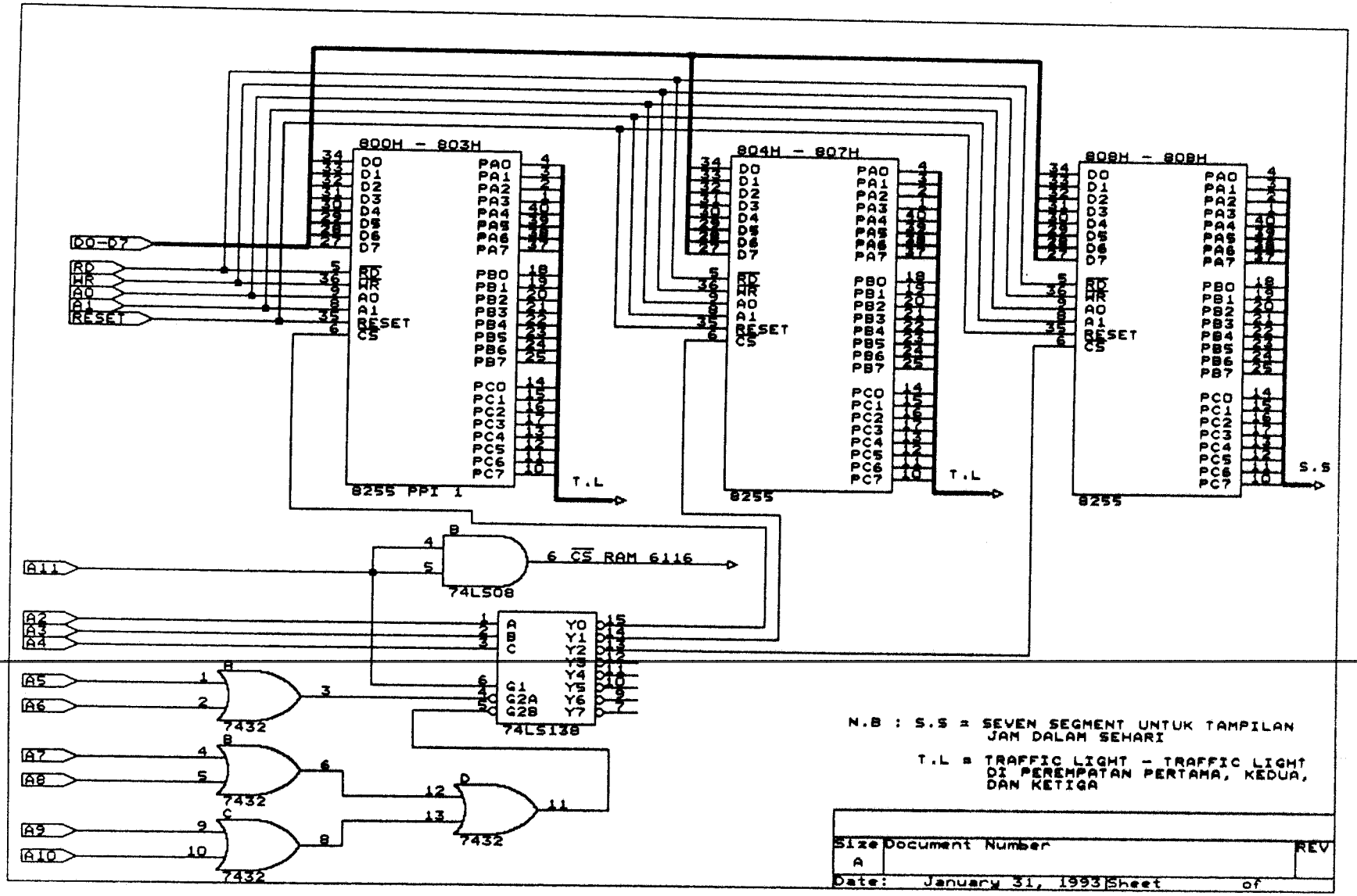
1. Arora, N.L., A Text book of Transportation Engineering, New India Publishing House, New Dehli, 1982.
2. Busono, Lebih Lanjut Tentang Peningkatan Daya Guna Komputer, P.T. Elex Media Komputindo, Jakarta, 1991.
3. Nashelsky, Louis, Introduction to Digital Computer Technology, second edition, USA, 1977.
4. Steeman, J.P.M., Data sheet Book, P.T. Elex Media Komputindo, Jakarta, 1989.
5. Wharton, John, An Introduction to the Intel MCS-51 TM Single-Chip Microcomputer Family, Intel Corporation, Santa Clara, 1980.
6. ----------. TTL Data Book, Fairchild Canera and Instrument Corporation, California, 1978.
7. ----------, Volume II, Intel Corp., Santa Clara California USA, 1986.

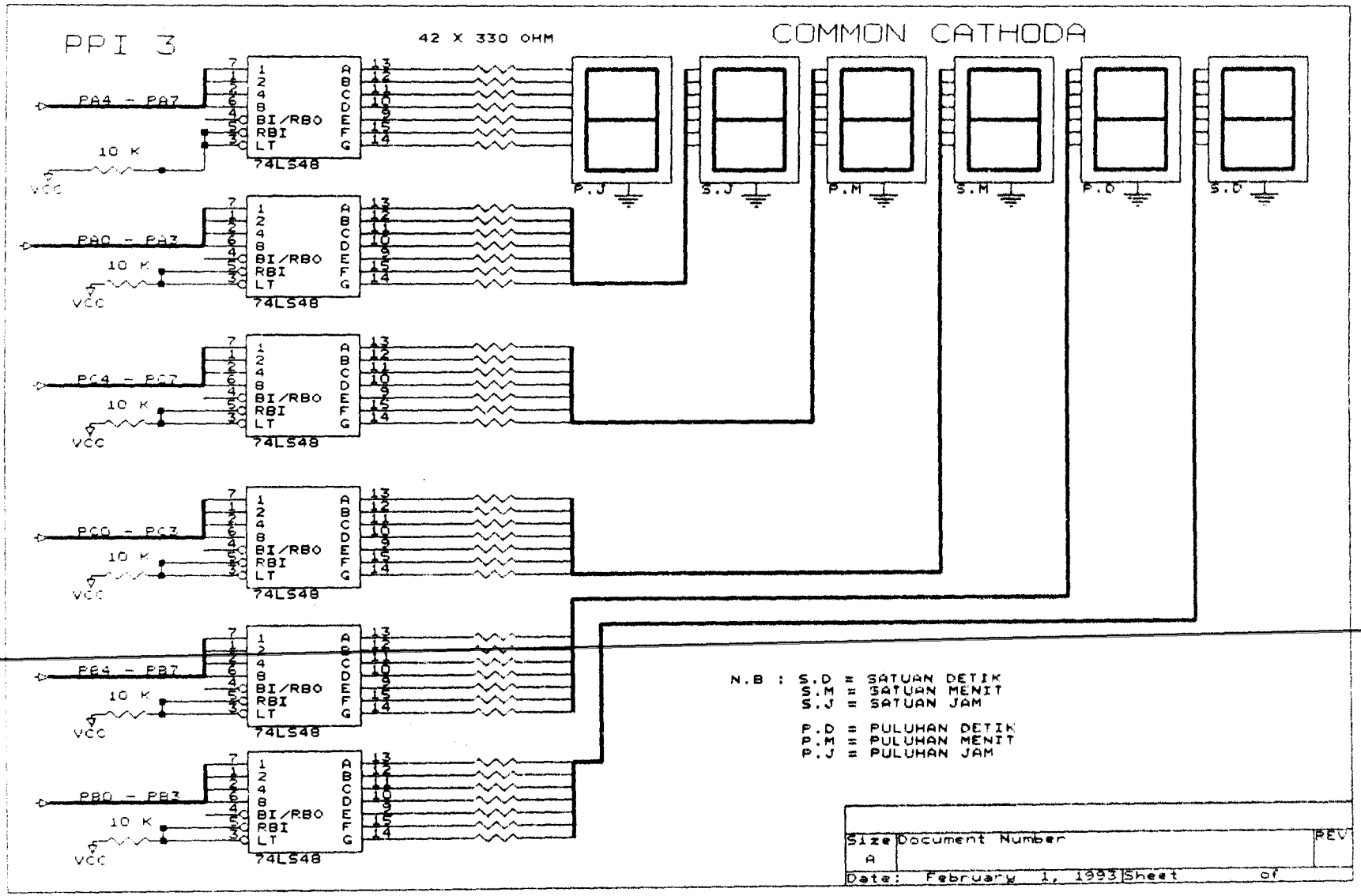

C
B
2
A
B
3
B
C
A
C
B
B
1
B
A



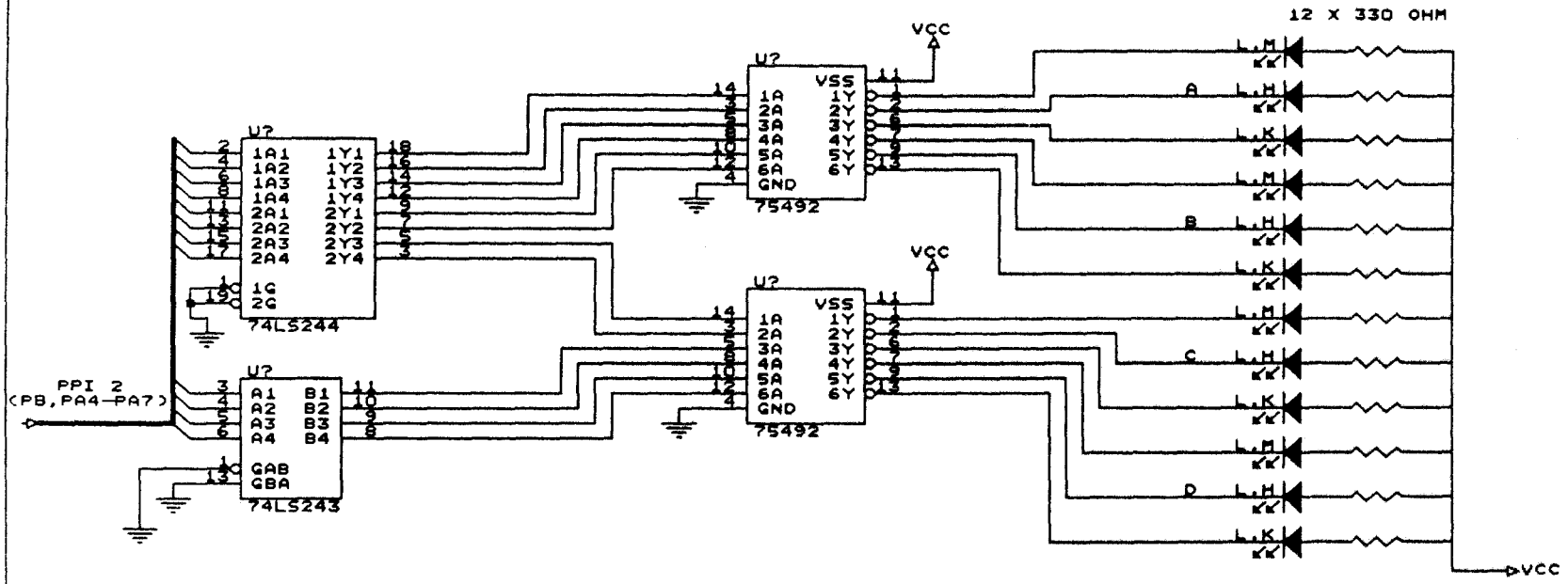
N.B : P1.3 = ADANYA KECELAKAAN DI PEREMPATAN PERTAMA
P1.4 = ADANYA KECELAKAAN DI PEREMPATAN KEDUA
P1.5 = ADANYA KECELAKAAN DI PEREMPATAN KETIGA
P1.6 = ADA KERETA API DI PEREMPATAN 2 DAN 3
P1.7 = START UNTUK SISTIM

Size	Document Number	REV
A		
Date: January 31, 1993	Sheet of	


8000H - 8003H
8255 PPI 1
8255
8255
T.L
T.L
S.S
D0-D7
RD
WR
A0
A1
RESET
CS
PA0
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
PC0
PC1
PC2
PC3
PC4
PC5
PC6
PC7
A11
6 CS RAM 6116
74LS08
A2
A3
A4
A5
A6
A7
A8
A9
A10
74LS138
7432
7432
7432
7432
N.B : S.S = SEVEN SEGMENT UNTUK TAMPILAN JAM DALAM SEHARI
T.L = TRAFFIC LIGHT - TRAFFIC LIGHT DI PEREMPATAN PERTAMA, KEDUA, DAN KETIGA
Size A
Document Number
REV
Date: January 31, 1993
Sheet of


PPI 3
42 X 330 OHM
COMMON CATHODA
PA4 - PA7
PA0 - PA3
PC4 - PC7
PC0 - PC3
PB4 - PB7
PB0 - PB3
10 K
VCC
BI/RBO
RBI
LT
74LS48
P.J
S.J
P.M
S.M
P.D
S.D
N.B : S.D = SATUAN DETIK
S.M = SATUAN MENIT
S.J = SATUAN JAM
P.D = PULUHAN DETIK
P.M = PULUHAN MENIT
P.J = PULUHAN JAM
Size
A
Document Number
REV
Date: February 1, 1993
Sheet
of

UNTUK PEREMPATAN PERTAMA



N.B: L.M = LAMPU MERAH

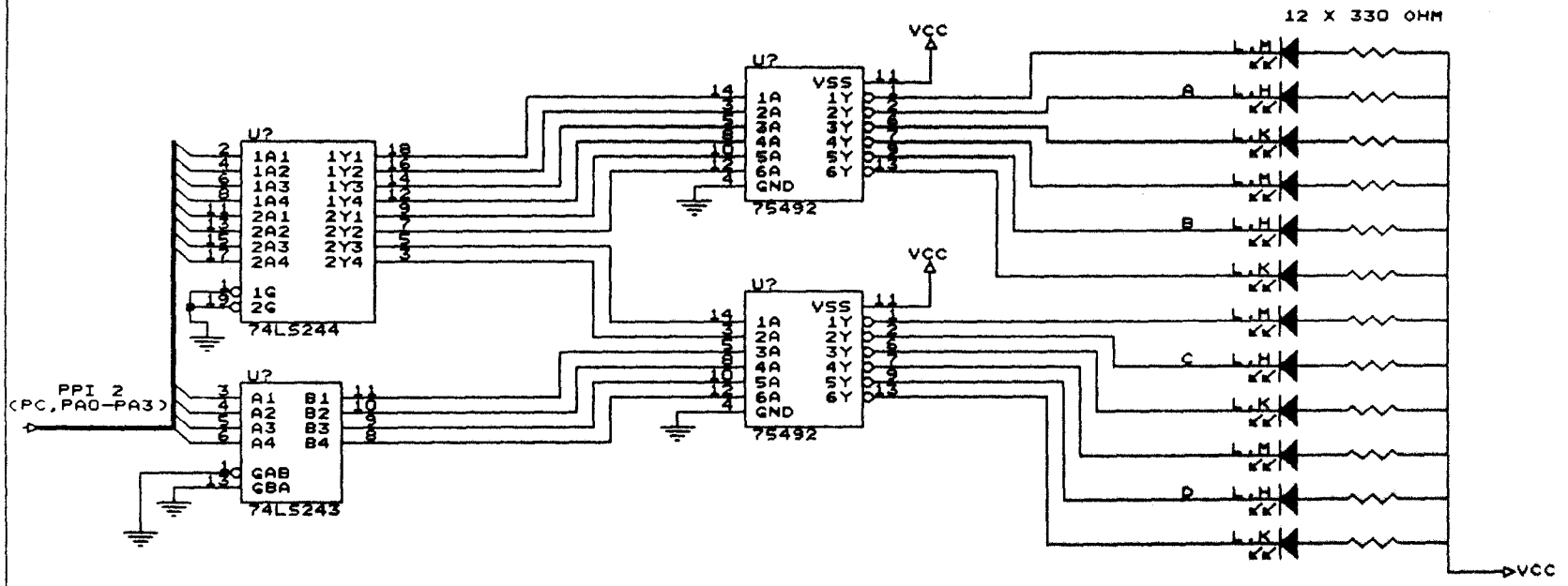
L.H = LAMPU HIJAU

L.K = LAMPU KUNING

A = PEREMPATAN 1 POSISI A
B = PEREMPATAN 1 POSISI B
C = PEREMPATAN 1 POSISI C
D = PEREMPATAN 1 POSISI D

Size	Document Number	REV
A		
Date: January 31, 1993	Sheet of	

UNTUK PEREMPATAN KEDUA



N.B: L.M = LAMPU MERAH

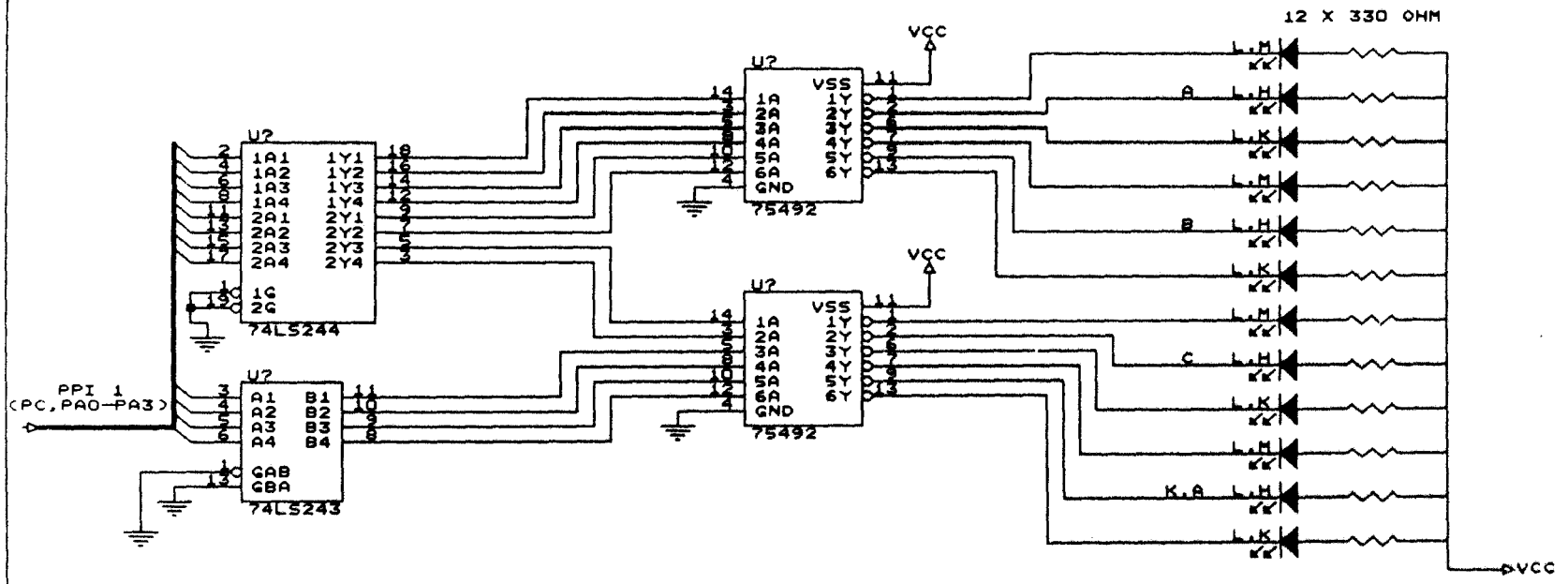
L.H = LAMPU HIJAU

L.K = LAMPU KUNING

A = PEREMPATAN 2 POSISI A
B = PEREMPATAN 2 POSISI B
C = PEREMPATAN 2 POSISI C
D = PEREMPATAN 2 POSISI D

Size	Document Number	REV
A		
Date: January 31, 1993	Sheet of	

UNTUK PEREMPATAN KETIGA



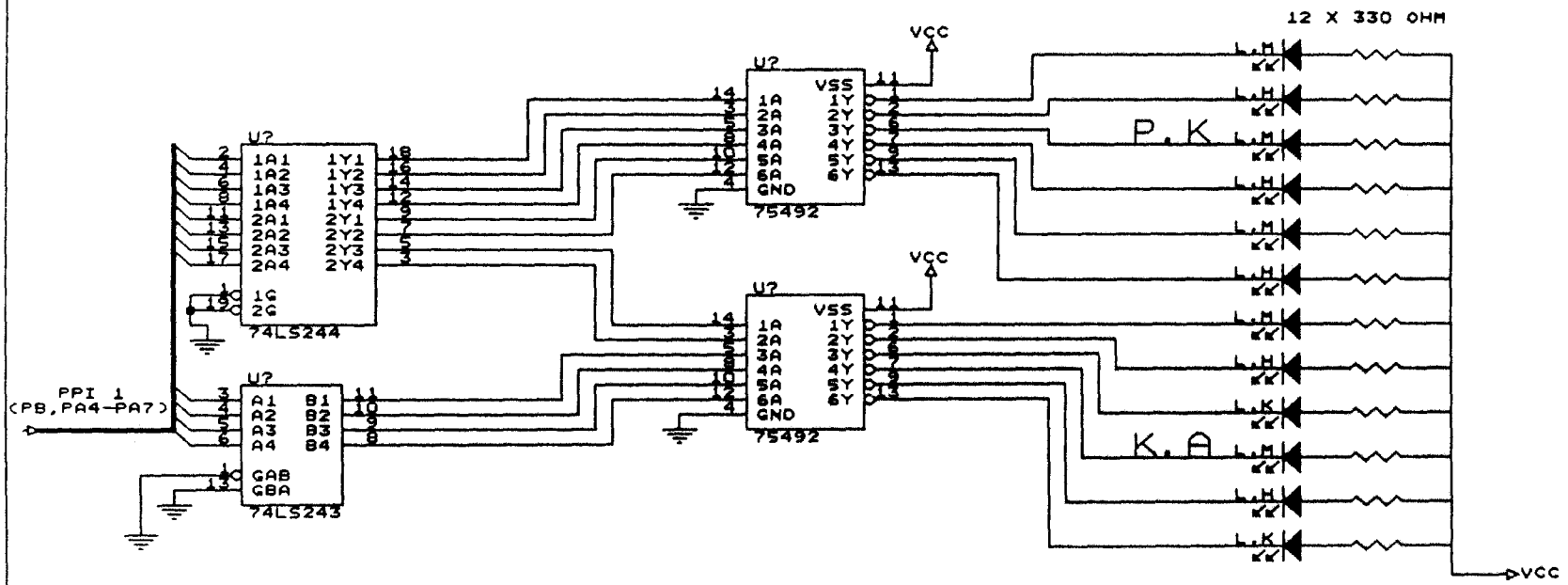
N.B: L.M = LAMPU MERAH

L.H = LAMPU HIJAU

L.K = LAMPU KUNING

A = PEREMPATAN 3 POSISI A
B = PEREMPATAN 3 POSISI B
C = PEREMPATAN 3 POSISI C
K.A = ARAH LALU LINTAS YANG MENUJU KE REL KERETA API

Size	Document Number	REV
A		
Date: January 31, 1993	Sheet of	



N.B: L.M = LAMPU MERAH

L.H = LAMPU HIJAU

L.K = LAMPU KUNING

K.A = ARAH LALU LINTAS YANG MENUJU KE REL KERETA API

P.K = LAMPU PEJALAN KAKI

Size	Document Number	REV
A		
Date: January 31, 1993	Sheet of	

```
;=============================================
;     SINKRONISASI TRAFFIC LIGHT
;
;     OLEH          : FELIX PHANDEAN
;     NRP           : 2882200986
;     FAKULTAS      : TEKNOLOGI INDUSTRI
;     JURUSAN       : TEKNIK ELEKTRO
;     BIDANG STUDI  : ELEKTRONIKA
;
;     INSTITUT TEKNOLOGI SEPULUH NOPEMBER
;                    SURABAYA
;=============================================
DELAY3 DATA 60H
DELAY4 DATA 61H
DELAY5 DATA 62H
DELAY6 DATA 63H
DELAY7 DATA 64H
DELAY8 DATA 65H
DELAY9 DATA 66H
DELAY10 DATA 67H

        ORG 0000H
        LJMP INIT
        ORG 0003H
        LJMP K_A
        ORG 001BH
        RETI
;-----------------------
;KOMUNIKASI SERIAL
;-----------------------
            ORG 0023H
            PUSH B
            MOV DPTR,#08H
            MOV R4,#26
SERIAL:
            JNB RI,$
            MOV A,SBUF          ;TERIMA DATA
            CLR RI              ;DARI KOMPUTER
            MOVX @DPTR,A
            INC DPTR
            DJNZ R4,SERIAL
            POP B
            RETI
;----------------------
;PROGRAM INISIALISASI
;----------------------
        ORG 50H
INIT:   MOV SP,#50H
        MOV TCON,#01000001B     ;TIMER1 MODE 2(serial)
        MOV TMOD,#00100000B
        MOV IE,#10010001B
        MOV IP,#00000001B
        MOV SCON,#01010000B     ;SERIAL PORT MODE 1
        MOV 87H,#10000000B
        MOV TH1,#243            ;BAUD RATE = 4800
        SETB P1.7
```

```
        CLR P1.6
;-----------------------
;INISIALISASI PPI 8255
;-----------------------
        MOV A,#10000000B
        MOV DPTR,#803H
        MOVX @DPTR,A
        MOV DPTR,#807H
        MOVX @DPTR,A
        MOV DPTR,#80BH
        MOVX @DPTR,A

        MOV A,#0                ;MATIKAN SEMUA LAMPU
        MOV DPTR,#800H          ;KE LED-LED
        MOVX @DPTR,A
        INC DPTR
        MOVX @DPTR,A
        INC DPTR
        MOVX @DPTR,A
        INC DPTR
        INC DPTR
        MOVX @DPTR,A
        INC DPTR
        MOVX @DPTR,A
        INC DPTR
        MOVX @DPTR,A
        MOV R2,#30H
        MOV R3,#32H
        MOV R4,#34H
        JB P1.7,$               ;JIKA P1.7=0 BERARTI START
             MOV DPTR,#1AH      ;ATAU PROGRAM DIJALANKAN
             MOVX A,@DPTR
             MOV B,#10
             DIV AB
             SWAP A
             ORL A,B
             MOV DPTR,#809H      ;TAMPILKAN ANGKA DETIK
             MOVX @DPTR,A        ;KE SEVEN SEGMENT
             MOV DPTR,#1BH
             MOVX A,@DPTR
             MOV B,#10
             DIV AB
             SWAP A
             ORL A,B
             MOV DPTR,#80AH      ;TAMPILKAN ANGKA MENIT
             MOVX @DPTR,A        ;KE SEVEN SEGMENT
             MOV DPTR,#1CH
             MOVX A,@DPTR
             MOV B,#10
             DIV AB
             SWAP A
             ORL A,B
             MOV DPTR,#808H        ;TAMPILKAN ANGKA JAM
             MOVX @DPTR,A          ;KE SEVEN SEGMENT
             LCALL CEK_JAM
```

```
        MOV DPTR,#805H
        MOV A,#01001010B
        MOVX @DPTR,A
        MOV R5,#38H             ;PEREMPATAN 1(A)
        MOV DPTR,#1DH           ;LAMPU HIJAU NYALA
        MOVX A,@DPTR
        CJNE A,#13,FP2B
        MOV DPTR,#800H          ;PEREMPATAN 2(C)
        MOV A,#00100000B        ;LAMPU HIJAU NYALA
        MOV R1,A
        MOVX @DPTR,A
        INC DPTR
        MOV A,#00100101B
        MOVX @DPTR,A
        MOV DPTR,#804H
        MOV A,#01000001B
        MOV B,A
        MOVX @DPTR,A
        MOV DPTR,#1EH
        MOVX A,@DPTR
        MOV DPTR,#51H
        MOVX @DPTR,A
        MOV R6,#44H
        AJMP ADA3
FP2B:   CJNE A,#12,FP2A
        MOV DPTR,#800H          ;PEREMPATAN 2(B)
        MOV A,#00010000B        ;LAMPU HIJAU NYALA
        MOV R1,A
        MOVX @DPTR,A
        INC DPTR
        MOV A,#01000101B
        MOVX @DPTR,A
        MOV DPTR,#804H
        MOV A,#01000010B
        MOV B,A
        MOVX @DPTR,A
        MOV DPTR,#1EH
        MOVX A,@DPTR
        MOV DPTR,#51H
        MOVX @DPTR,A
        MOV R6,#42H
        AJMP ADA3
FP2A:   MOV DPTR,#800H          ;PEREMPATAN 2(A)
        MOV A,#10010000B        ;LAMPU HIJAU NYALA
        MOV R1,A
        MOVX @DPTR,A
        INC DPTR
        MOV A,#00101000B
        MOVX @DPTR,A
        MOV DPTR,#804H
        MOV A,#01000001B
        MOV B,A
        MOVX @DPTR,A
        MOV DPTR,#1EH
        MOVX A,@DPTR
```

```
        MOV DPTR,#51H
        MOVX @DPTR,A
        MOV R6,#40H
ADA3:   MOV DPTR,#1FH
        MOVX A,@DPTR
        CJNE A,#18,FP3B
        MOV DPTR,#800H          ;PEREMPATAN 3(C)
        MOV A,R1                ;LAMPU HIJAU NYALA
        ORL A,#00001010B
        MOV R1,A
        MOVX @DPTR,A
        MOV DPTR,#802H
        MOV A,#00100100B
        MOVX @DPTR,A
        MOV DPTR,#806H
        MOV A,#01010101B
        MOV R0,A
        MOVX @DPTR,A
        MOV DPTR,#20H
        MOVX A,@DPTR
        MOV DPTR,#50H
        MOVX @DPTR,A
        MOV R7,#4CH
        AJMP STABIL
FP3B:   CJNE A,#17,FP3A
        MOV DPTR,#800H          ;PEREMPATAN 3(B)
        MOV A,R1                ;LAMPU HIJAU NYALA
        ORL A,#00000001B
        MOV R1,A
        MOVX @DPTR,A
        MOV DPTR,#802H
        MOV A,#01000101B
        MOVX @DPTR,A
        MOV DPTR,#804H
        MOV A,B
        ORL A,#00001000B
        MOV B,A
        MOVX @DPTR,A
        MOV DPTR,#806H
        MOV A,#01010100B
        MOV R0,A
        MOVX @DPTR,A
        MOV DPTR,#20H
        MOVX A,@DPTR
        MOV DPTR,#50H
        MOVX @DPTR,A
        MOV R7,#4AH
        AJMP STABIL
FP3A:   MOV DPTR,#800H          ;PEREMPATAN 3(A)
        MOV A,R1                ;LAMPU HIJAU NYALA
        ORL A,#00001001B
        MOV R1,A
        MOVX @DPTR,A
        MOV DPTR,#802H
        MOV A,#00101000B
```

```
                MOVX @DPTR,A
                MOV DPTR,#804H
                MOV A,B
                ORL A,#00001000B
                MOV B,A
                MOVX @DPTR,A
                MOV DPTR,#806H
                MOV A,#01010100B
                MOV R0,A
                MOVX @DPTR,A
                MOV DPTR,#20H
                MOVX A,@DPTR
                MOV DPTR,#50H
                MOVX @DPTR,A
                MOV R7,#48H
STABIL:
                JB P1.6,YOS1
WAN1:           JNB P1.3,FP302
                JNB P1.4,WK306
                JNB P1.5,WK310
                LJMP FP40F
YOS1:           JNB P1.3,FP302
                JNB P1.4,WK306
                JNB P1.5,WK310
                LJMP YOS2
WK306:          LJMP FP306
WK310:          LJMP FP310
FP302:          CJNE R5,#38H,K1A        ;KECELAKAAN PADA
                INC R2                  ;PEREMPATAN 1(A)
                CJNE R2,#31H,MATI1
                MOV DPTR,#804H
                MOV A,B
                ANL A,#10001111B
                ORL A,#10000000B
                MOV B,A
                MOVX @DPTR,A
                INC DPTR
                MOV A,#01001100B
                MOVX @DPTR,A
                AJMP FP300
MATI1:          MOV DPTR,#804H
                MOV A,B
                ANL A,#00001111B
                MOV B,A
                MOVX @DPTR,A
                INC DPTR
                MOV A,#01001000B
                MOVX @DPTR,A
                MOV R2,#30H
                AJMP FP300
K1A:            CJNE R5,#39H,K1B
                MOV DPTR,#804H
                MOV A,B
                ANL A,#10001111B
                ORL A,#10000000B
```

```
                MOV B,A
                MOVX @DPTR,A
                AJMP FP300
K1B:            CJNE R5,#3AH,K1C        ;KECELAKAAN PADA
                INC R2                  ;PEREMPATAN 1(B)
                CJNE R2,#31H,MATI2
                MOV DPTR,#804H
                MOV A,B
                ANL A,#10001111B
                ORL A,#10000000B
                MOV B,A
                MOVX @DPTR,A
                INC DPTR
                MOV A,#01100001B
                MOVX @DPTR,A
                AJMP FP300
MATI2:          MOV DPTR,#804H
                MOV A,B
                ANL A,#00001111B
                MOV B,A
                MOVX @DPTR,A
                INC DPTR
                MOV A,#01000001B
                MOVX @DPTR,A
                MOV R2,#30H
                AJMP FP300
K1C:            CJNE R5,#3BH,K1D
                AJMP FP300
K1D:            CJNE R5,#3CH,FP300      ;KECELAKAAN PADA
                INC R2                  ;PEREMPATAN 1(C)
                CJNE R2,#31H,MATI3
                MOV DPTR,#804H
                MOV A,B
                ANL A,#00111111B
                ORL A,#00110000B
                MOV B,A
                MOVX @DPTR,A
                INC DPTR
                MOV A,#00001001B
                MOVX @DPTR,A
                AJMP FP300
MATI3:          MOV DPTR,#804H
                MOV A,B
                ANL A,#00101111B
                MOV B,A
                MOVX @DPTR,A
                INC DPTR
                MOV A,#00001001B
                MOVX @DPTR,A
                MOV R2,#30H
FP300:          JNB P1.4,FP306
                AJMP FP304
FP306:          CJNE R6,#40H,K2A        ;KECELAKAAN PADA
                INC R3                  ;PEREMPATAN 2(A)
                CJNE R3,#33H,MATI4
```

```
                MOV DPTR,#801H
                MOV A,#00110000B
                MOVX @DPTR,A
                AJMP FP304
MATI4:          MOV DPTR,#801H
                MOV A,#00100000B
                MOVX @DPTR,A
                MOV R3,#32H
                AJMP FP304
K2A:            CJNE R6,#41H,K2B
                AJMP FP304
K2B:            CJNE R6,#42H,K2C
                INC R3                      ;KECELAKAAN PADA
                CJNE R3,#33H,MATI5          ;PEREMPATAN 2(B)
                MOV DPTR,#801H
                MOV A,#10000110B
                MOVX @DPTR,A
                JB P1.6,YUL1
                MOV DPTR,#804H
                MOV A,B
                ANL A,#11111100B
                ORL A,#00000100B
                MOV B,A
                MOVX @DPTR,A
YUL1:           AJMP FP304
MATI5:          MOV DPTR,#801H
                MOV A,#00000100B
                MOVX @DPTR,A
                JB P1.6,YUL1AB
                MOV DPTR,#804H
                MOV A,B
                ANL A,#11111000B
                MOV B,A
                MOVX @DPTR,A
YUL1AB:         MOV R3,#32H
                AJMP FP304
K2C:            CJNE R6,#43H,K2D
                MOV DPTR,#801H
                MOV A,#10000110B
                MOVX @DPTR,A
                AJMP FP304
K2D:            CJNE R6,#44H,FP304          ;KECELAKAAN PADA
                INC R3                      ;PEREMPATAN 2(C)
                CJNE R3,#33H,MATI6
                MOV DPTR,#800H
                MOV A,R1
                ANL A,#01001111B
                ORL A,#01000000B
                MOV R1,A
                MOVX @DPTR,A
                INC DPTR
                MOV A,#00100110B
                MOVX @DPTR,A
                AJMP FP304
MATI6:          MOV DPTR,#800H
```

```
        MOV A,R1
        ANL A,#00001111B
        MOV R1,A
        MOVX @DPTR,A
        INC DPTR
        MOV A,#00100100B
        MOVX @DPTR,A
        MOV R3,#32H
FP304:  JNB P1.5,FP310
        AJMP YOS2F
FP310:  CJNE R7,#48H,K3A
        INC R4
        CJNE R4,#35H,MATI7      ;KECELAKAAN PADA
        MOV DPTR,#802H          ;PEREMPATAN 3(A)
        MOV A,#00110000B
        MOVX @DPTR,A
        AJMP YOS2F
MATI7:  MOV DPTR,#802H
        MOV A,#00100000B
        MOVX @DPTR,A
        MOV R4,#34H
        AJMP YOS2F
K3A:    CJNE R7,#49H,K3B
        AJMP YOS2F
K3B:    CJNE R7,#4AH,K3C
        INC R4                  ;KECELAKAAN PADA
        CJNE R4,#35H,MATI8      ;PEREMPATAN 3(B)
        JB P1.6,YUL2
        MOV DPTR,#802H
        MOV A,#10000110B
        MOVX @DPTR,A
        AJMP YOS2F
YUL2:   MOV DPTR,#802H
        MOV A,#10000100B
        MOVX @DPTR,A
        AJMP YOS2F
MATI8:  MOV DPTR,#802H
        MOV A,#00000100B
        MOVX @DPTR,A
        MOV R4,#34H
        AJMP YOS2F
K3C:    CJNE R7,#4BH,K3D
        AJMP YOS2F
K3D:    CJNE R7,#4CH,YUL3       ;KECELAKAAN PADA
        INC R4                  ;PEREMPATAN 3(C)
        CJNE R4,#35H,MATI9
        MOV DPTR,#800H
        MOV A,R1
        ANL A,#11111100B
        ORL A,#00001100B
        MOV R1,A
        MOVX @DPTR,A
        JB P1.6,YUL3
        MOV DPTR,#806H
        MOV A,R0
```

```
                ANL A,#11111110B
                ORL A,#00000010B
                MOV R0,A
                MOVX @DPTR,A
YUL3:           AJMP YOS2F
MATI9:          MOV DPTR,#800H
                MOV A,R1
                ANL A,#11111000B
                MOV R1,A
                MOVX @DPTR,A
                JB P1.6,YUL3AB
                MOV DPTR,#806H
                MOV A,R0
                ANL A,#11111100B
                MOV R0,A
                MOVX @DPTR,A
YUL3AB:         MOV R4,#34H
                AJMP YOS2F
YOS2:           LCALL HIT1
                JB P1.6,YUL20          ;ADA KERETA API
                CJNE R6,#42H,YUL7      ;YANG LEWAT DI ANTARA
                MOV DPTR,#804H         ;PEREMPATAN KEDUA DAN
                MOV A,B                ;KETIGA
                ANL A,#11111010B
                ORL A,#00000010B
                MOV B,A
                MOVX @DPTR,A
                AJMP YUL8
YUL20:          LJMP YOS10
YUL7:           CJNE R6,#43H,YUL8
                MOV DPTR,#804H
                MOV A,B
                ANL A,#11111100B
                ORL A,#00000100B
                MOV B,A
                MOVX @DPTR,A
YUL8:           CJNE R7,#4AH,YUL10
                MOV DPTR,#800H
                MOV A,R1
                ANL A,#11110111B
                MOV R1,A
                MOVX @DPTR,A
                MOV DPTR,#802H
                MOV A,#01000101B
                MOVX @DPTR,A
                AJMP YOS10
YUL10:          CJNE R7,#4BH,YUL11
                MOV DPTR,#800H
                MOV A,R1
                ANL A,#11110111B
                MOV R1,A
                MOVX @DPTR,A
                MOV DPTR,#802H
                MOV A,#10000110B
                MOVX @DPTR,A
```

```
        AJMP YOS10
YUL11:  CJNE R7,#4CH,YUL13
        MOV DPTR,#804H
        MOV A,B
        ANL A,#11110111B
        MOV B,A
        MOVX @DPTR,A
        MOV DPTR,#806H
        MOV A,R0
        ANL A,#11111101B
        ORL A,#01000001B
        MOV R0,A
        MOVX @DPTR,A
        AJMP YOS10
YUL13:  CJNE R7,#4DH,RUT1
        MOV DPTR,#804H
        MOV A,B
        ANL A,#11110111B
        MOV B,A
        MOVX @DPTR,A
        MOV DPTR,#806H
        MOV A,R0
        ANL A,#11111110B
        ORL A,#01000010B
        MOV R0,A
        MOVX @DPTR,A
RUT1:   AJMP YOS10
YOS2F:  LCALL HIT1
        JB P1.6,LAN9
        CJNE R6,#42H,LAN7
        MOV DPTR,#804H
        MOV A,B
        ANL A,#11111010B
        ORL A,#00000010B
        MOV B,A
        MOVX @DPTR,A
        AJMP LAN8
LAN9:   AJMP RUT10
LAN7:   CJNE R6,#43H,LAN8
        MOV DPTR,#804H
        MOV A,B
        ANL A,#11111100B
        ORL A,#00000100B
        MOV B,A
        MOVX @DPTR,A
LAN8:   CJNE R7,#4AH,LAN10
        MOV DPTR,#800H
        MOV A,R1
        ANL A,#11110111B
        MOV R1,A
        MOVX @DPTR,A
        MOV DPTR,#802H
        MOV A,#01000101B
        MOVX @DPTR,A
        AJMP RUT10
```

```
LAN10:    CJNE R7,#4BH,TAN11
          MOV DPTR,#800H
          MOV A,R1
          ANL A,#11110111B
          MOV R1,A
          MOVX @DPTR,A
          MOV DPTR,#802H
          MOV A,#10000110B
          MOVX @DPTR,A
          AJMP RUT10
TAN11:    CJNE R7,#4CH,TAN13
          MOV DPTR,#804H
          MOV A,B
          ANL A,#11110111B
          MOV B,A
          MOVX @DPTR,A
          MOV DPTR,#806H
          MOV A,R0
          ANL A,#11111101B
          ORL A,#01000001B
          MOV R0,A
          MOVX @DPTR,A
          AJMP RUT10
TAN13:    CJNE R7,#4DH,RUT10
          MOV DPTR,#804H
          MOV A,B
          ANL A,#11110111B
          MOV B,A
          MOVX @DPTR,A
          MOV DPTR,#806H
          MOV A,R0
          ANL A,#11111110B
          ORL A,#01000010B
          MOV R0,A
          MOVX @DPTR,A
RUT10:      JB P1.3,FP802
            JB P1.4,PH10
            JB P1.5,PH11
            LJMP YOS10
PH10:       LJMP FP806
PH11:       LJMP FP810
FP802:       CJNE R5,#38H,KL199      ;KECALAKAAN DI
             MOV DPTR,#804H          ;PEREMPATAN PERTAMA
             MOV A,B                 ;TELAH TERATASI
             ANL A,#01001111B
             ORL A,#01000000B        ;LAMPU HIJAU
             MOV B,A                 ;MENYALA DI 1(A)
             MOVX @DPTR,A            ;NORMAL KEMBALI
             INC DPTR
             MOV A,#01001010B
             MOVX @DPTR,A
             AJMP FP804
KL199:       CJNE R5,#39H,KL1        ;LAMPU KUNING DI 1(A)
             MOV DPTR,#804H          ;NORMAL KEMBALI
             MOV A,B
```

```
        ANL A,#01001111B
        ORL A,#01000000B
        MOV B,A
        MOVX @DPTR,A
        AJMP FP804
KL1:    CJNE R5,#3AH,KL2
        MOV DPTR,#804H          ;LAMPU HIJAU DI 1(B)
        MOV A,B                 ;NORMAL KEMBALI
        ANL A,#01001111B
        ORL A,#01000000B
        MOV B,A
        MOVX @DPTR,A
        INC DPTR
        MOV A,#01010001B
        MOVX @DPTR,A
        AJMP FP804
KL2:    CJNE R5,#3BH,KL299      ;LAMPU KUNING DI 1(B)
        AJMP FP804              ;NORMAL KEMBALI
KL299:  CJNE R5,#3CH,FP804
        MOV DPTR,#804H          ;LAMPU HIJAU DI 1(C)
        MOV A,B                 ;NORMAL KEMBALI
        ANL A,#00101111B
        ORL A,#00100000B
        MOV B,A
        MOVX @DPTR,A
        INC DPTR
        MOV A,#10001001B
        MOVX @DPTR,A
FP804:  JB P1.4,FP806           ;LAMPU KUNING DI 1(C)
        AJMP FP808              ;NORMAL KEMABLI
FP806:  CJNE R6,#40H,KM199
        MOV DPTR,#801H          ;KECELAKAAN DI
        MOV A,#00101000B        ;PEREMPATAN KEDUA
        MOVX @DPTR,A            ;TELAH TERATASI
        AJMP FP808              ;LAMPU HIJAU DI 2(A)
KM199:  CJNE R6,#41H,KM1        ;NORMAL KEMBALI
        AJMP FP808              ;LAMPU KUNING DI 2(A)
KM1:    CJNE R6,#42H,KM2        ;NORMAL KEMBALI
        MOV DPTR,#801H
        MOV A,#01000101B
        MOVX @DPTR,A            ;LAMPU HIJAU DI 2(B)
        JB P1.6,YUL4            ;NORMAL KEMBALI
        MOV DPTR,#804H
        MOV A,B
        ANL A,#11111010B
        ORL A,#00000010B
        MOV B,A
        MOVX @DPTR,A
YUL4:   AJMP FP808
KM2:    CJNE R6,#43H,KM299
        MOV DPTR,#801H          ;LAMPU KUNING DI 2(B)
        MOV A,#10000101B        ;NORMAL KEMBALI
        MOVX @DPTR,A
        AJMP FP808
KM299:  CJNE R6,#44H,FP808
```

```
                MOV DPTR,#800H          ;LAMPU HIJAU DI 2(C)
                MOV A,R1                ;NORMAL KEMBALI
                ANL A,#00101111B
                ORL A,#00100000B
                MOV R1,A
                MOVX @DPTR,A
                INC DPTR
                MOV A,#00100101B
                MOVX @DPTR,A
FP808:          JB P1.5,FP810           ;LAMPU KUNING DI 2(C)
                AJMP YOS10
FP810:          CJNE R7,#48H,KN199      ;KECALAKAAN DI
                MOV DPTR,#802H          ;PEREMPATAN KETIGA
                MOV A,#00101000B        ;TELAH TERATASI
                MOVX @DPTR,A            ;LAMPU HIJAU DI 3(A)
                AJMP YOS10              ;NORMAL KEMBALI
KN199:          CJNE R7,#49H,KN1        ;LAMPU KUNING DI 3(A)
                AJMP YOS10              ;NORMAL KEMBALI
KN1:            CJNE R7,#4AH,KN2
                JB P1.6,YUL5            ;LAMPU HIJAU DI 3(B)
                 MOV DPTR,#802H         ;NORMAL KEMBALI
                MOV A,#01000101B
                MOVX @DPTR,A
                AJMP YOS10
YUL5:            MOV DPTR,#802H
                MOV A,#01000100B
                MOVX @DPTR,A
                AJMP YOS10
KN2:            CJNE R7,#4BH,KN299      ;LAMPU KUNING DI 3(B)
                AJMP YOS10              ;NORMAL KEMBALI
KN299:          CJNE R7,#4CH,FP409F
                MOV DPTR,#800H          ;LAMPU HIJAU DI 3(C)
                MOV A,R1                ;NORMAL KEMBALI
                ANL A,#11111010B
                ORL A,#00001010B
                MOV R1,A
                MOVX @DPTR,A
                JB P1.6,FP409F
                MOV DPTR,#806H
                MOV A,R0
                ANL A,#11111101B
                ORL A,#01000001B
                MOV R0,A
                MOVX @DPTR,A
FP409F:         AJMP YOS10              ;LAMPU KUNING DI 3(C)
FP40F:      LCALL HIT1                  ;NORMAL KEMBALI
YOS10:      MOV DPTR,#52H
            MOVX A,@DPTR
            DEC A
            MOVX @DPTR,A
            JNZ FP41
            LCALL NYALA1
            MOV A,DELAY7                ;SUDAH SATU SIKLUS
            CJNE A,#22h,FP41            ;ATAU BELUM, KALAU
            LCALL CEK_JAM               ;SUDAH KE CEK_JAM
```

```
FP41:     MOV DPTR,#51H
          MOVX A,@DPTR
          DEC A
          MOVX @DPTR,A
          JNZ FP42
          LCALL NYALA2
FP42:     MOV DPTR,#50H
          MOVX A,@DPTR
          DEC A
          MOVX @DPTR,A
          JNZ FP43
          LCALL NYALA3
FP43:     LJMP STABIL
;-----------------------------------
;PROGRAM PROSEDUR HITUNG 1 DETIK
;-----------------------------------
HIT1:     MOV DELAY6,#31
FEP1:     MOV DELAY5,#63
FEP2:     MOV DELAY4,#254
          DJNZ DELAY4,$
          DJNZ DELAY5,FEP2
          DJNZ DELAY6,FEP1
          MOV DPTR,#1AH
          MOVX A,@DPTR
          INC A                    ;INCREMENT DETIK
          CJNE A,#60,WK10
          MOV A,#0
          MOVX @DPTR,A
          MOV DPTR,#809H
          MOVX @DPTR,A
          MOV DPTR,#1BH
          MOVX A,@DPTR
          INC A                    ;INCREMENT MENIT
          CJNE A,#60,WK11
          MOV A,#0H
          MOVX @DPTR,A
          MOV DPTR,#80AH
          MOVX @DPTR,A
          MOV DPTR,#1CH
          MOVX A,@DPTR
          INC A
          CJNE A,#24,WK12
          MOV A,#0                 ;INCREMENT JAM
          MOVX @DPTR,A
          MOV DPTR,#808H
          MOVX @DPTR,A
          AJMP FP40M
WK10:     MOVX @DPTR,A
          PUSH B
          MOV B,#10
          DIV AB
          SWAP A
          ORL A,B
          MOV DPTR,#809H           ;TAMPILKAN ANGKA DETIK
          MOVX @DPTR,A             ;KE SEVEN SEGMENT
```

```
            POP B
            AJMP FP40M
WK11:       MOVX @DPTR,A
            PUSH B
            MOV B,#10
            DIV AB
            SWAP A
            ORL A,B
            MOV DPTR,#80AH          ;TAMPILKAN ANGKA MENIT
            MOVX @DPTR,A            ;KE SEVEN SEGMENT
            POP B
            AJMP FP40M
WK12:       MOVX @DPTR,A
            PUSH B
            MOV B,#10
            DIV AB
            SWAP A
            ORL A,B
            MOV DPTR,#808H          ;TAMPILKAN ANGKA JAM
            MOVX @DPTR,A            ;KE SEVEN SEGMENT
            POP B
FP40M:      RET
;---------------------
; PROSEDUR CEK JAM
;---------------------
CEK_JAM:
TERAS:  MOV DPTR,#1CH           ;JAM DICEK
        MOVX A,@DPTR
        CJNE A,#5,FP1
        AJMP STATE1
FP1:    JC PAN3
        CJNE A,#7,FP2
        AJMP STATE2
PAN3:   AJMP STATE3
FP2:    JC STATE1
        CJNE A,#9,FP3
        AJMP STATE1
FP3:    JC STATE2
        CJNE A,#12,FP4
        AJMP STATE2
FP4:    JC STATE1
        CJNE A,#14,FP5
        AJMP STATE1
FP5:    JC STATE2
        CJNE A,#16,FP6
        AJMP STATE2
FP6:    JC STATE1
        CJNE A,#18,FP7
        AJMP STATE1
FP7:    JC STATE2
        CJNE A,#23,FP8
        AJMP STATE3
FP8:    JC STATE1
        AJMP STATE3
```

```
STATE1:   MOV DPTR,#19H
          MOVX A,@DPTR          ;JAM BIASA
          MOV DPTR,#52H
          MOVX @DPTR,A
          MOV DELAY3,#75H
          MOV DELAY7,#25H
          AJMP FP9
STATE2:   MOV DPTR,#10H
          MOVX A,@DPTR
          MOV DPTR,#52H         ;JAM SIBUK
          MOVX @DPTR,A
          MOV DELAY3,#77H
          MOV DELAY7,#25H
          AJMP FP9
STATE3:   MOV DPTR,#800H        ;JAM SEPI, LAMPU KUNING
          MOV A,#01000100B      ;NYALA DAN MATI SELAMA
          MOVX @DPTR,A          ;1 DETIK
          INC DPTR
          MOV A,#10010010B
          MOVX @DPTR,A          ;SEMUA LAMPU KUNING NYALA
          INC DPTR
          MOV A,#10010010B
          MOVX @DPTR,A
          INC DPTR
          INC DPTR
          MOV A,#10010100B
          MOVX @DPTR,A
          INC DPTR
          MOV A,#00100100B
          MOVX @DPTR,A
          INC DPTR
          MOV A,#00000010B
          MOVX @DPTR,A
          LCALL HIT1
          MOV DPTR,#800H        ;SEMUA LAMPU MATI
          MOV A,#0
          MOVX @DPTR,A
          INC DPTR
          MOVX @DPTR,A
          INC DPTR
          MOVX @DPTR,A
          INC DPTR
          INC DPTR
          MOVX @DPTR,A
          INC DPTR
          MOVX @DPTR,A
          INC DPTR
          MOVX @DPTR,A
          LCALL HIT1
          AJMP TERAS
FP9:      RET
;-----------------------
; INTERRUPT KERETA API
;-----------------------
K_A:      PUSH PSW
```

```
        MOV DPTR,#800H
        MOV A,R1
        ORL A,#00001000B
        MOV R1,A
        MOVX @DPTR,A
        CJNE R7,#4AH,YOS21
        MOV DPTR,#802H
        MOV A,#01000100B
        MOVX @DPTR,A
        AJMP YOS22
YOS21:  CJNE R7,#4BH,YOS22
        MOV DPTR,#802H
        MOV A,#10000100B
        MOVX @DPTR,A
YOS22:  MOV DPTR,#804H
        MOV A,B
        ANL A,#11111001B
        ORL A,#00001001B
        MOV B,A
        MOVX @DPTR,A
        MOV DPTR,#806H
        MOV A,R0
        ANL A,#11111100B
        MOV R0,A
        MOVX @DPTR,A
        POP PSW
        RETI
;----------------------------
;PROGRAM PROSEDUR NYALA1
;----------------------------
NYALA1:
            INC R5
MAM1:       CJNE R5,#38H,KO199
            MOV DPTR,#804H          ;LAMPU HIJAU DI 1(A)
            MOV A,B                 ;MENYALA
            ANL A,#01001111B
            ORL A,#01000000B
            MOV B,A
            MOVX @DPTR,A
            INC DPTR
            MOV A,#01001010B
            MOVX @DPTR,A
            MOV DPTR,#806H
            MOV A,R0
            ANL A,#11110111B
            ORL A,#00000100B
            MOV R0,A
            MOVX @DPTR,A
            MOV A,DELAY3
            CJNE A,#75H,SIBUK1
            MOV DPTR,#19H
            MOVX A,@DPTR            ;JAM BIASA
            MOV DPTR,#52H
            MOVX @DPTR,A
            LJMP KO10
```

```
SIBUK1:     MOV DPTR,#10H
            MOVX A,@DPTR
            MOV DPTR,#52H           ;JAM SIBUK
            MOVX @DPTR,A
            MOV DELAY7,#22H
            LJMP KO10
KO199:      CJNE R5,#39H,KO1
            MOV DPTR,#805H          ;LAMPU KUNING DI 1(A)
            MOV A,#01001100B        ;NYALA
            MOVX @DPTR,A
            MOV A,#3
            MOV DPTR,#52H
            MOVX @DPTR,A
            LJMP KO10
KO1:        CJNE R5,#3AH,KO2
            MOV DPTR,#805H          ;LAMPU HIJAU DI 1(B)
            MOV A,#01010001B        ;NYALA
            MOVX @DPTR,A
            MOV A,DELAY3
            CJNE A,#75H,SIBUK2
            MOV DPTR,#18H           ;JAM BIASA
            MOVX A,@DPTR
            MOV DPTR,#52H
            MOVX @DPTR,A
            LJMP KO10
SIBUK2:     MOV DPTR,#0FH
            MOVX A,@DPTR            ;JAM SIBUK
            MOV DPTR,#52H
            MOVX @DPTR,A
            LJMP KO10
KO2:        CJNE R5,#3BH,KO299
            MOV DPTR,#804H          ;LAMPU KUNING DI 1(B)
            MOV A,B                 ;NYALA
            ANL A,#10001111B
            ORL A,#10000000B
            MOV B,A
            MOVX @DPTR,A
            INC DPTR
            MOV A,#01100001B
            MOVX @DPTR,A
            MOV A,#3
            MOV DPTR,#52H
            MOVX @DPTR,A
            LJMP KO10
KO299:      CJNE R5,#3CH,KO3
            MOV DPTR,#804H          ;LAMPU HIJAU DI 1(C)
            MOV A,B                 ;NYALA
            ANL A,#00101111B
            ORL A,#00100000B
            MOV B,A
            MOVX @DPTR,A
            INC DPTR
            MOV A,#10001001B
            MOVX @DPTR,A
            MOV A,DELAY3
```

```
                CJNE A,#75H,SIBUK3
                MOV DPTR,#17H           ;JAM BIASA
                MOVX A,@DPTR
                MOV DPTR,#52H
                MOVX @DPTR,A
                LJMP KO10
SIBUK3:         MOV DPTR,#0EH           ;JAM SIBUK
                MOVX A,@DPTR
                MOV DPTR,#52H
                MOVX @DPTR,A
                LJMP KO10
KO3:            CJNE R5,#3DH,KO4
                MOV DPTR,#804H          ;LAMPU KUNING DI 1(C)
                MOV A,B                 ;NYALA
                ORL A,#00110000B
                MOV B,A
                MOVX @DPTR,A
                INC DPTR
                MOV A,#00001001B
                MOVX @DPTR,A
                MOV A,#3
                MOV DPTR,#52H
                MOVX @DPTR,A
                LJMP KO10
KO4:            CJNE R5,#3EH,KO5
                MOV DPTR,#804H          ;ALL RED
                MOV A,B
                ORL A,#00100000B
                MOV B,A
                MOVX @DPTR,A
                INC DPTR
                MOV A,#01001001B
                MOVX @DPTR,A
                MOV A,#1
                MOV DPTR,#52H
                MOVX @DPTR,A
                LJMP KO10
KO5:            CJNE R5,#3FH,KO6
                MOV DPTR,#21H
                MOVX A,@DPTR
                JZ KO9
                MOV DPTR,#52H           ;LAMPU HIJAU
                MOVX @DPTR,A            ;PEJALAN KAKI NYALA
                MOV DPTR,#804H
                MOV A,B
                ANL A,#00101111B
                MOV B,A
                MOVX @DPTR,A
                INC DPTR
                MOV A,#01001001B
                MOVX @DPTR,A
                INC DPTR
                MOV A,R0
                ANL A,#11111011B
                ORL A,#00001000B
```

```
                MOV R0,A
                MOVX @DPTR,A
                AJMP KO10
KO9:            MOV R5,#37H
                INC R5
                LJMP MAM1
KO6:            MOV DPTR,#806H
                MOV A,R0                    ;ALL RED
                ANL A,#11110111B
                ORL A,#00000100B
                MOV R0,A
                MOVX @DPTR,A
                MOV A,#1
                MOV DPTR,#52H
                MOVX @DPTR,A
                MOV R5,#37H
KO10:           RET
;-----------------------------
;PROGRAM PROSEDUR NYALA2
;-----------------------------
NYALA2:
                INC R6
MAM2:           CJNE R6,#40H,KA199
                MOV DPTR,#800H              ;LAMPU HIJAU DI 2(A)
                MOV A,R1                    ;NYALA
                ANL A,#10011111B
                ORL A,#10010000B
                MOV R1,A
                MOVX @DPTR,A
                INC DPTR
                MOV A,#00101000B
                MOVX @DPTR,A
                MOV DPTR,#806H
                MOV A,R0
                ANL A,#11011111B
                ORL A,#00010000B
                MOV R0,A
                MOVX @DPTR,A
                MOV A,DELAY3
                CJNE A,#75H,SIBUK4
                MOV DPTR,#16H
                MOVX A,@DPTR                ;JAM BIASA
                MOV DPTR,#51H
                MOVX @DPTR,A
                LJMP KA10
SIBUK4:         MOV DPTR,#0DH
                MOVX A,@DPTR
                MOV DPTR,#51H               ;JAM SIBUK
                MOVX @DPTR,A
                LJMP KA10
KA199:          CJNE R6,#41H,KA1
                MOV DPTR,#801H              ;LAMPU KUNING DI 2(A)
                MOV A,#00110000B            ;NYALA
                MOVX @DPTR,A
                MOV A,#3
```

```
                MOV DPTR,#51H
                MOVX @DPTR,A
                LJMP KA10
KA1:            CJNE R6,#42H,KA2
                MOV DPTR,#800H          ;LAMPU HIJAU DI 2(B)
                MOV A,R1                ;NYALA
                ANL A,#00011111B
                MOV R1,A
                MOVX @DPTR,A
                INC DPTR
                MOV A,#01000101B
                MOVX @DPTR,A
                JNB P1.6,NAK2
                MOV DPTR,#804H
                MOV A,B
                ANL A,#11111001B
                ORL A,#00000001B
                MOV B,A
                MOVX @DPTR,A
                AJMP PAPB
NAK2:           MOV DPTR,#804H
                MOV A,B
                ANL A,#11111010B
                ORL A,#00000010B
                MOV B,A
                MOVX @DPTR,A
PAPB:           MOV A,DELAY3
                CJNE A,#75H,SIBUK5
                MOV DPTR,#15H           ;JAM BIASA
                MOVX A,@DPTR
                MOV DPTR,#51H
                MOVX @DPTR,A
                LJMP KA10
SIBUK5:         MOV DPTR,#0CH
                MOVX A,@DPTR            ;JAM SIBUK
                MOV DPTR,#51H
                MOVX @DPTR,A
                LJMP KA10
KA2:            CJNE R6,#43H,KA299
                MOV DPTR,#801H          ;LAMPU KUNING DI 2(B)
                MOV A,#10000101B        ;NYALA
                MOVX @DPTR,A
                JNB P1.6,NAK3
                MOV DPTR,#804H
                MOV A,B
                ANL A,#11111001B
                ORL A,#00000001B
                MOV B,A
                MOVX @DPTR,A
                AJMP PAPC
NAK3:           MOV DPTR,#804H
                MOV A,B
                ANL A,#11111100B
                ORL A,#00000100B
                MOV B,A
```

```
                MOVX @DPTR,A
PAPC:           MOV A,#3
                MOV DPTR,#51H
                MOVX @DPTR,A
                LJMP KA10
KA299:          CJNE R6,#44H,KA3
                MOV DPTR,#800H          ;LAMPU HIJAU DI 2(C)
                MOV A,R1                ;NYALA
                ANL A,#00101111B
                ORL A,#00100000B
                MOV R1,A
                MOVX @DPTR,A
                INC DPTR
                MOV A,#00100101B
                MOVX @DPTR,A
                MOV DPTR,#804H
                MOV A,B
                ANL A,#11111001B
                ORL A,#00000001B
                MOV B,A
                MOVX @DPTR,A
                MOV A,DELAY3
                CJNE A,#75H,SIBUK6
                MOV DPTR,#14H           ;JAM BIASA
                MOVX A,@DPTR
                MOV DPTR,#51H
                MOVX @DPTR,A
                LJMP KA10
SIBUK6:         MOV DPTR,#0BH
                MOVX A,@DPTR            ;JAM SIBUK
                MOV DPTR,#51H
                MOVX @DPTR,A
                LJMP KA10
KA3:            CJNE R6,#45H,KA4
                MOV DPTR,#800H          ;LAMPU KUNING DI 2(C)
                MOV A,R1                ;NYALA
                ANL A,#01001111B
                ORL A,#01000000B
                MOV R1,A
                MOVX @DPTR,A
                INC DPTR
                MOV A,#00100110B
                MOVX @DPTR,A
                MOV A,#3
                MOV DPTR,#51H
                MOVX @DPTR,A
                LJMP KA10
KA4:            CJNE R6,#46H,KA5        ;ALL RED
                MOV DPTR,#800H
                MOV A,R1
                ANL A,#10011111B
                ORL A,#10010000B
                MOV R1,A
                MOVX @DPTR,A
                INC DPTR
```

```
                MOV A,#00100100B
                MOVX @DPTR,A
                MOV A,#1
                MOV DPTR,#51H
                MOVX @DPTR,A
                LJMP KA10
KA5:            CJNE R6,#47H,KA6
                MOV DPTR,#21H
                MOVX A,@DPTR
                JZ KA9
                MOV DPTR,#51H           ;LAMPU HIJAU
                MOVX @DPTR,A            ;PEJALAN KAKI NYALA
                MOV DPTR,#800H
                MOV A,R1
                ANL A,#10011111B
                ORL A,#10010000B
                MOV R1,A
                MOVX @DPTR,A
                INC DPTR
                MOV A,#00100100B
                MOVX @DPTR,A
                MOV DPTR,#806H
                MOV A,R0
                ANL A,#11101111B
                ORL A,#00100000B
                MOV R0,A
                MOVX @DPTR,A
                AJMP KA10
KA9:            MOV R6,#3FH
                INC R6
                LJMP MAM2
KA6:            MOV DPTR,#806H
                MOV A,R0                ;ALL RED
                ANL A,#11011111B
                ORL A,#00010000B
                MOV R0,A
                MOVX @DPTR,A
                MOV A,#1
                MOV DPTR,#52H
                MOVX @DPTR,A
                MOV R6,#3FH
KA10:           RET

;-----------------------------
;PROGRAM PROSEDUR NYALA3
;-----------------------------
NYALA3:
                INC R7
MAM3:           CJNE R7,#48H,KB199
                MOV DPTR,#800H          ;LAMPU HIJAU DI 3(A)
                MOV A,R1                ;NYALA
                ANL A,#11111001B
                ORL A,#00001001B
                MOV R1,A
                MOVX @DPTR,A
```

```
                MOV DPTR,#802H
                MOV A,#00101000B
                MOVX @DPTR,A
                MOV DPTR,#804H
                MOV A,B
                ORL A,#00001000B
                MOV B,A
                MOVX @DPTR,A
                MOV DPTR,#806H
                MOV A,R0
                ANL A,#01111100B
                ORL A,#01000000B
                MOV R0,A
                MOVX @DPTR,A
YOS91:          MOV A,DELAY3
                CJNE A,#75H,SIBUK7
                MOV DPTR,#13H
                MOVX A,@DPTR            ;JAM BIASA
                MOV DPTR,#50H
                MOVX @DPTR,A
                LJMP KB10
SIBUK7:         MOV DPTR,#0AH
                MOVX A,@DPTR            ;JAM SIBUK
                MOV DPTR,#50H
                MOVX @DPTR,A
                LJMP KB10
KB199:          CJNE R7,#49H,KB1
                MOV DPTR,#802H          ;LAMPU KUNING DI 3(A)
                MOV A,#00110000B        ;NYALA
                MOVX @DPTR,A
                MOV A,#3
                MOV DPTR,#50H
                MOVX @DPTR,A
                LJMP KB10
KB1:            CJNE R7,#4AH,KB2
                JNB P1.6,NAK4
                MOV DPTR,#800H          ;LAMPU HIJAU DI 3(B)
                MOV A,R1
                ANL A,#11111001B
                ORL A,#00001001B
                MOV R1,A
                MOVX @DPTR,A
                MOV DPTR,#802H
                MOV A,#01000100B
                MOVX @DPTR,A
                AJMP PAPD
NAK4:           MOV DPTR,#800H
                MOV A,R1
                ANL A,#11110001B
                MOV R1,A
                MOVX @DPTR,A
                MOV DPTR,#802H
                MOV A,#01000101B
                MOVX @DPTR,A
PAPD:           MOV A,DELAY3
```

```
                CJNE A,#75H,SIBUK8
                MOV DPTR,#12H           ;JAM BIASA
                MOVX A,@DPTR
                MOV DPTR,#50H
                MOVX @DPTR,A
                LJMP KB10
SIBUK8:         MOV DPTR,#09H
                MOVX A,@DPTR            ;JAM SIBUK
                MOV DPTR,#50H
                MOVX @DPTR,A
                LJMP KB10
KB2:            CJNE R7,#4BH,KB299
                JNB P1.6,NAK5
                MOV DPTR,#800H          ;LAMPU KUNING DI 3(B)
                MOV A,R1                ;NYALA
                ANL A,#11111001B
                ORL A,#00001001B
                MOV R1,A
                MOVX @DPTR,A
                MOV DPTR,#802H
                MOV A,#10000100B
                MOVX @DPTR,A
                AJMP PAPE
NAK5:           MOV DPTR,#802H
                MOV A,#10000110B
                MOVX @DPTR,A
PAPE:           MOV A,#3
                MOV DPTR,#50H
                MOVX @DPTR,A
                LJMP KB10
KB299:          CJNE R7,#4CH,KB3
                MOV DPTR,#800H          ;LAMPU HIJAU DI 3(C)
                MOV A,R1                ;NYALA
                ANL A,#11111010B
                ORL A,#00001010B
                MOV R1,A
                MOVX @DPTR,A
                INC DPTR
                INC DPTR
                MOV A,#00100100B
                MOVX @DPTR,A
                JNB P1.6,NAK6
                MOV DPTR,#804H
                MOV A,B
                ORL A,#00001000B
                MOV B,A
                MOVX @DPTR,A
                MOV DPTR,#806H
                MOV A,R0
                ANL A,#01111100B
                ORL A,#01000000B
                MOV R0,A
                MOVX @DPTR,A
                AJMP PAPE
NAK6:           MOV DPTR,#804H
```

```
                MOV A,B
                ANL A,#11110111B
                MOV B,A
                MOVX @DPTR,A
                MOV DPTR,#806H
                MOV A,R0
                ANL A,#01111101B
                ORL A,#01000001B
                MOV R0,A
                MOVX @DPTR,A
PAPF:           MOV A,DELAY3
                CJNE A,#75H,SIBUK9
                MOV DPTR,#11H
                MOVX A,@DPTR            ;JAM BIASA
                MOV DPTR,#50H
                MOVX @DPTR,A
                LJMP KB10
SIBUK9:         MOV DPTR,#08H
                MOVX A,@DPTR            ;JAM SIBUK
                MOV DPTR,#50H
                MOVX @DPTR,A
                LJMP KB10
KB3:            CJNE R7,#4DH,KB4
                MOV DPTR,#800H          ;LAMPU KUNING DI 3(C)
                MOV A,R1                ;NYALA
                ANL A,#11111100B
                ORL A,#00001100B
                MOV R1,A
                MOVX @DPTR,A
                JNB P1.6,NAK7
                MOV DPTR,#804H
                MOV A,B
                ORL A,#00001000B
                MOV B,A
                MOVX @DPTR,A
                MOV DPTR,#806H
                MOV A,R0
                ANL A,#01111100B
                ORL A,#01000000B
                MOV R0,A
                MOVX @DPTR,A
                AJMP PAPG
NAK7:           MOV DPTR,#806H
                MOV A,R0
                ANL A,#01111110B
                ORL A,#01000010B
                MOV R0,A
                MOVX @DPTR,A
PAPG:           MOV A,#3
                MOV DPTR,#50H
                MOVX @DPTR,A
                LJMP KB10
KB4:            CJNE R7,#4EH,KB5        ;ALL RED
                MOV DPTR,#800H
                MOV A,R1
```

```
        ANL A,#11111001B
        ORL A,#00001001B
        MOV R1,A
        MOVX @DPTR,A
        MOV DPTR,#804H
        MOV A,B
        ORL A,#00001000B
        MOV B,A
        MOVX @DPTR,A
        MOV DPTR,#806H
        MOV A,R0
        ANL A,#01111100B
        ORL A,#01000000B
        MOV R0,A
        MOVX @DPTR,A
        MOV A,#1
        MOV DPTR,#50H
        MOVX @DPTR,A
        LJMP KB10
KB4:    CJNE R7,#4FH,KB6
        MOV DPTR,#21H
        MOVX A,@DPTR
        JZ KB9
        MOV DPTR,#50H           ;LAMPU HIJAU
        MOVX @DPTR,A            ;PEJALAN KAKI NYALA
        MOV DPTR,#800H
        MOV A,R1
        ANL A,#11111001B
        ORL A,#00001001B
        MOV R1,A
        MOVX @DPTR,A
        MOV DPTR,#804H
        MOV A,B
        ORL A,#00001000B
        MOV B,A
        MOVX @DPTR,A
        MOV DPTR,#806H
        MOV A,R0
        ANL A,#10111100B
        ORL A,#10000000B
        MOV R0,A
        MOVX @DPTR,A
        AJMP KB10
KB9:    MOV R7,#47H
        INC R7
        LJMP MAM3
KB6:    MOV DPTR,#806H          ;ALL RED
        MOV A,R0
        ANL A,#01111111B
        ORL A,#01000000B
        MOV R0,A
        MOVX @DPTR,A
        MOV A,#1
        MOV DPTR,#50H
        MOVX @DPTR,A
```

```
                MOV R7,#47H
KB10:           RET

                END
```

```
uses crt,dos;

const
     PORT = 0;
     WT = 5;

type
   pixel        = record
                    karakter : byte;
                    atribut : byte;
                  end;
   bufferlayar = array[1..25,1..80] of pixel;
var
   S1, S2, V1, TD, SD, AT : real;
   I_HL1A, I_HL1B, I_HL1C, I_HL2A, I_HL2B, I_HL2C, I_HL3A,
   I_HL3B, I_HL3C, PK, II_HL1A, II_HL1B, II_HL1C, II_HL2A,
   II_HL2B, II_HL2C, II_HL3A, II_HL3B, II_HL3C,
   Tjam, Tmnt, Tdtk, FP2, TFP2, FP3, TFP3 : byte;
   T1, T2, TE : word;
   status_1, status_2, status_3 : boolean;

procedure KembaliLayarSemula(kolom1,baris1,kolom2,baris2 : byte;
          var layarsimpan : bufferlayar);
var
   posisi:word;
   i,j    :byte;
begin
   for i:=baris1 to baris2 do
   begin
      for j:=kolom1 to kolom2 do
         with layarsimpan[i,j] do
         begin
            posisi := (((i-1)*80+j)-1)*2;
            mem[$B800:$0000+posisi] := ord(karakter);
            mem[$B800:$0000+posisi+1] := atribut;
         end;
   end;
end;

procedure SimpanLayar(kolom1,baris1,kolom2,baris2 : byte;
          var layarsimpan : bufferlayar);
var
   posisi:word;
   i,j    :byte;
begin
   for i:=baris1 to baris2 do
   begin
      for j:=kolom1 to kolom2 do
         with layarsimpan[i,j] do
         begin
            posisi := (((i-1)*80+j)-1)*2;
```

```
            karakter := mem[$B800:$0000+posisi];
            atribut := mem[$B800:$0000+posisi+1];
          end;
    end;
end;

function getb(n : byte):byte;
var
    tekan : char;
    str   : string;
    i     : byte;
    kode,hasil  : integer;
begin
    i := 1;
    while i<=n do
    begin
       str[i] := readkey;
       case str[i] of
         #8 : begin
                if i>1 then
                begin
                  write(#8#32#8); dec(i);
                end;
              end;
         #13: begin
                str[0] := chr(i-1);
                val(str,hasil,kode);
                getb := hasil;
                exit;
              end;
         #0 : ;
         else begin
                gotoxy(wherex,wherey); write(str[i]);
                inc(i);
              end;
       end;
    end;
    str[0] := chr(n);
    val(str,hasil,kode);
    getb := hasil;
end;

procedure RubahAtribut(kolom1,baris1,kolom2,baris2,atribut:byte);
var
    posisi : word;
    i,j    : byte;
begin
    for I:=baris1 to baris2 do
    begin
       for j:=kolom1 to kolom2 do
       begin
```

```
            posisi:=(((I-1)*80+j)-1)*2;
            mem[$b800:$0000+posisi+1]:=atribut;
         end;
      end;
end;

procedure box(kolom1,baris1,kolom2,baris2,atribut : byte);
var
   i,j : byte;
begin
   textattr := atribut;
   for i:=baris1 to baris2 do
   begin
      gotoxy(kolom1,i); write('3');
      gotoxy(kolom2,i); write('3');
      for j:=kolom1+1 to kolom2-1 do
      begin
         gotoxy(j,i); write(' ');
      end;
   end;
   for i:=kolom1 to kolom2 do
   begin
      gotoxy(i,baris1); write('D');
      gotoxy(i,baris2); write('D');
   end;
   gotoxy(kolom1,baris1); write('Z');
   gotoxy(kolom2,baris1); write('?');
   gotoxy(kolom1,baris2); write('@');
   gotoxy(kolom2,baris2); write('Y');
end;

Function menu : byte;
var
   buffer_1 : bufferlayar;
   tekan    : char;
   pilihan  : byte;
begin
   box(30,9,48,15,$17);
   textattr := $1E;
   gotoxy(33,10); write('INISIALISASI ');
   gotoxy(33,11); write('  JAM BIASA  ');
   gotoxy(33,12); write('  JAM SIBUK  ');
   gotoxy(33,13); write('TRANSFER DATA');
   gotoxy(33,14); write('   KELUAR    ');
   Simpanlayar(30,9,48,15,buffer_1);
   textattr := $3F;
   pilihan := 1;
   rubahatribut(31,10,47,10,$3F);
   repeat
      tekan := readkey;
      case tekan of
```

```
        #0 : begin
               tekan := readkey;
               case tekan of
                  #80 : begin
                          inc(pilihan);
                          if pilihan > 5 then pilihan := 1;
                        end;
                  #72 : begin
                          dec(pilihan);
                          if pilihan < 1 then pilihan := 5;
                        end;
               end;
             end;
        #27: begin
               menu := 10;
               exit;
             end;
      end;
      kembalilayarsemula(30,9,48,15,buffer_1);
      rubahatribut(31,9+pilihan,47,9+pilihan,$3F)
   until tekan = #13;
   menu := pilihan;
end;


Procedure Inisialisasi;
label
  Inisial_1;
var
  YT : char;
begin
  box(12,5,68,16,$17);
  textattr := $1E;
  status_1 := TRUE;
Inisial_1 :
  repeat
  gotoxy(14,6);
  write('Jarak perempatan pertama dan kedua  (m)     :  ');
  READLN(S1); gotoxy(60,6); WRITE(S1:7:2);
  until S1>0;
  repeat
  gotoxy(14,7);
  write('Jarak perempatan kedua dan ketiga  (m)      :  ');
  READLN(S2); gotoxy(60,7); WRITE(S2:7:2);
  until S2>0;
  repeat
  gotoxy(14,8);
  write('Kecepatan stabil yang diinginkan (km/jam)   :  ');
  READLN(V1); gotoxy(60,8); WRITE(V1:7:2);
  until V1>0;
  repeat
```

```
    IF S1>61.38 THEN
    BEGIN
      T1 := ROUND(TD + ((3.6*(S1-SD))/V1));
      T2 := ROUND(3.6*(S2/V1));
    END
    ELSE BEGIN
      AT := V1/TD;
      T1 := ROUND(SQRT((2*S1*3.6)/AT));
      IF (S1 + S2)>61.38 THEN
      BEGIN
        TE := ROUND(TD + ((3.6*(S1 + S2 - SD))/V1));
        T2 := TE - T1;
      END
      ELSE
      BEGIN
        TE := ROUND(SQRT((2*(S2+S1)*3.6)/AT));
        T2 := TE - T1;
      END
    END
  END
  ELSE
  BEGIN
    IF V1<=50 THEN
    BEGIN
      TD := 10.98;
      SD := (10*V1*TD)/72;
      IF S1>76.25 THEN
      BEGIN
        T1 := ROUND(TD + ((3.6*(S1-SD))/V1));
        T2 := ROUND(3.6*(S2/V1));
      END
      ELSE BEGIN
        AT := V1/TD;
        T1 := ROUND(SQRT((2*S1*3.6)/AT));
        IF (S1 + S2)>76.25 THEN
        BEGIN
          TE := ROUND(TD + ((3.6*(S1 + S2 - SD))/V1));
          T2 := TE - T1;
        END
        ELSE
        BEGIN
          TE := ROUND(SQRT((2*(S2+S1)*3.6)/AT));
          T2 := TE - T1;
        END
      END
    END
    ELSE
    BEGIN
      IF V1<=55 THEN
      BEGIN
        TD := 12.66;
```

```
    SD := (10*V1*TD)/72;
    IF S1>96.71 THEN
    BEGIN
      T1 := ROUND(TD + ((3.6*(S1-SD))/V1));
      T2 := ROUND(3.6*(S2/V1));
    END
    ELSE BEGIN
      AT := V1/TD;
      T1 := ROUND(SQRT((2*S1*3.6)/AT));
      IF (S1 + S2)>96.71 THEN
      BEGIN
        TE := ROUND(TD + ((3.6*(S1 + S2 - SD))/V1));
        T2 := TE - T1;
      END
      ELSE
      BEGIN
        TE := ROUND(SQRT((2*(S2+S1)*3.6)/AT));
        T2 := TE - T1;
      END
    END
  END
  ELSE
  BEGIN
    IF V1<=60 THEN
    BEGIN
      TD := 14.66;
      SD := (10*V1*TD)/72;
      IF S1>122.17 THEN
      BEGIN
        T1 := ROUND(TD + ((3.6*(S1-SD))/V1));
        T2 := ROUND(3.6*(S2/V1));
      END
      ELSE BEGIN
        AT := V1/TD;
        T1 := ROUND(SQRT((2*S1*3.6)/AT));
        IF (S1 + S2)>122.17 THEN
        BEGIN
          TE := ROUND(TD + ((3.6*(S1 + S2 - SD))/V1));
          T2 := TE - T1;
        END
        ELSE
        BEGIN
          TE := ROUND(SQRT((2*(S2+S1)*3.6)/AT));
          T2 := TE - T1;
        END
      END
    END
    ELSE
    BEGIN
      TD := 14.66;
      SD := (10*V1*TD)/72;
```

```
                IF S1>122.17 THEN
                BEGIN
                  T1 := ROUND(TD + ((3.6*(S1-SD))/V1));
                  T2 := ROUND(3.6*(S2/V1));
                END
                ELSE BEGIN
                  AT := V1/TD;
                  T1 := ROUND(SQRT((2*S1*3.6)/AT));
                  IF (S1 + S2)>122.17 THEN
                  BEGIN
                    TE := ROUND(TD + ((3.6*(S1 + S2 - SD))/V1));
                    T2 := TE - T1;
                  END
                  ELSE
                  BEGIN
                    TE := ROUND(SQRT((2*(S2+S1)*3.6)/AT));
                    T2 := TE - T1;
                  END
                END;
              END;
            END;
          END;
        END;
      END;

      gotoxy(16,13); write('T1 = ',T1:3,' detik');
      gotoxy(34,13); write('T2 = ',T2:3,' detik');
      gotoxy(53,13); write('T = ',Tjam:2,':',Tmnt:2,':',Tdtk:2);
      rubahatribut(13,13,67,13,$3F);

      repeat
         gotoxy(14,15); write('Apakah data sudah benar (Y/T) ? ');
         readln(YT);
      until YT in ['Y','y','T','t'];
      case upcase(YT) of
        'Y' : exit;
        'T' : goto Inisial_1;
      end;
end;

Function Submenu_1 : byte;
var
   buffer_1 : bufferlayar;
   tekan    : char;
   pilihan  : byte;
begin
   box(6,7,75,17,$17);
   textattr := $1E;
   gotoxy(8,8);
   write('Lama lampu hijau perempatan 1(A) a menyala (detik) :',
         I_HL1A:3);
```

```
    gotoxy(8,9);
    write('Lama lampu hijau perempatan I(B) menyala (detik)   :',
          I_HL1B:3);
    gotoxy(8,10);
    write('Lama lampu hijau perempatan II(B) menyala (detik)  :',
          I_HL2B:3);
    gotoxy(8,11);
    write('Lama lampu hijau perempatan III(B) menyala (detik) :',
          I_HL3B:3);
    gotoxy(8,12);
    write('Data jam biasa sudah siap, kembali ke menu utama');
    gotoxy(8,14); write('I_HL1A = ',I_HL1A:3);
    gotoxy(8,15); write('I_HL1B = ',I_HL1B:3);
    gotoxy(8,16); write('I_HL1C = ',I_HL1C:3);
    gotoxy(30,14); write('I_HL2A = ',I_HL2A:3);
    gotoxy(30,15); write('I_HL2B = ',I_HL2B:3);
    gotoxy(30,16); write('I_HL2C = ',I_HL2C:3);
    gotoxy(52,14); write('I_HL3A = ',I_HL3A:3);
    gotoxy(52,15); write('I_HL3B = ',I_HL3B:3);
    gotoxy(52,16); write('I_HL3C = ',I_HL3C:3);

    Simpanlayar(6,7,75,17,buffer_1);
    textattr := $3F;
    pilihan := 1;
    rubahatribut(7,7+pilihan,74,7+pilihan,$3F);
    repeat
       tekan := readkey;
       case tekan of
         #0 : begin
                tekan := readkey;
                case tekan of
                   #80 : begin
                           inc(pilihan);
                           if pilihan > 5 then pilihan := 1;
                         end;
                   #72 : begin
                           dec(pilihan);
                           if pilihan < 1 then pilihan := 5;
                         end;
                end;
              end;
         #27: begin
                submenu_1 := 10;
                exit;
              end;
       end;
       kembalilayarsemula(6,7,75,17,buffer_1);
       rubahatribut(7,7+pilihan,74,7+pilihan,$3F)
    until tekan = #13;
    submenu_1 := pilihan;
end;
```

```
Procedure Biasa;
LABEL
     AWAL, AKHIR;
var
  temp : byte;
begin
  status_2 := TRUE;
  repeat
     temp := SubMenu_1;
     case temp of
        1 : begin
              repeat
                gotoxy(68,8); I_HL1A := getb(3);
              until I_HL1A > 0;
            end;
        2 : begin
              repeat
                gotoxy(68,9); I_HL1B := getb(3);
              until I_HL1B > 0;
                I_HL2A := I_HL1A + I_HL1B + WT;
            end;
        3 : begin
              repeat
                gotoxy(68,10); I_HL2B := getb(3);
              until I_HL2B > 0;
                I_HL3A := I_HL2A + I_HL2B + WT;
            end;
        4 : begin
              repeat
                gotoxy(68,11); I_HL3B := getb(3);
              until I_HL3B > 0;
                I_HL3C := I_HL3B - 5;
                I_HL2C := I_HL3B + I_HL3C + WT;
                I_HL1C := I_HL2B + I_HL2C + WT;

IF TJAM<7 THEN GOTO AWAL;
IF TJAM<9 THEN GOTO AKHIR;
IF TJAM<12 THEN GOTO AWAL;
IF TJAM<14 THEN GOTO AKHIR;
IF TJAM<16 THEN GOTO AWAL;
IF TJAM<18 THEN GOTO AKHIR;

AWAL:
IF T1<=(I_HL1C - WT) THEN
BEGIN
  IF (T1 + I_HL2B)<=(I_HL1C - WT) THEN
  BEGIN
    FP2 := 13;
    TFP2 := T1;
  END
  ELSE
```

```
  BEGIN
    FP2 := 12;
    TFP2 := I_HL2B + T1 - I_HL1C + WT;
  END
END
ELSE
BEGIN
  IF (T1 - I_HL1C+WT)>I_HL2A THEN
  BEGIN
   IF (T1 - I_HL1C+WT - I_HL2A)>I_HL2C THEN
   BEGIN
    IF (T1 - I_HL1C+WT - I_HL2A - I_HL2C)>I_HL2B THEN
    BEGIN
     IF (T1 - I_HL1C+WT - I_HL2A - I_HL2C - I_HL2B)>I_HL2A THEN
     BEGIN
      IF (T1 - I_HL1C+WT-2*I_HL2A-I_HL2C-I_HL2B)>I_HL2C THEN
      BEGIN
       IF (T1- I_HL1C+WT-2*I_HL2A-2*I_HL2C-I_HL2B)>I_HL2 THEN
       BEGIN
        IF (T1-I_HL1C+WT-2*I_HL2A-2*I_HL2C-2*I_HL2B)>I_HL2A THEN
        BEGIN
          FP2 := 13;
          TFP2 := T1 - I_HL1C+WT-3*I_HL2A-2*I_HL2C-2*I_HL2B;
        END
        ELSE
        BEGIN
          FP2 := 11;
          TFP2 := T1 - I_HL1C+WT-2*I_HL2A-2*I_HL2C-2*I_HL2B;
        END
       END
       ELSE
       BEGIN
         FP2 := 12;
         TFP2 := T1 - I_HL1C+WT-2*I_HL2A-2*I_HL2C-I_HL2B;
       END
      END
      ELSE
      BEGIN
        FP2 := 13;
        TFP2 := T1 - I_HL1C + WT - 2*I_HL2A - I_HL2C - I_HL2B;
      END
     END
     ELSE
     BEGIN
       FP2 := 11;
       TFP2 := T1 - I_HL1C + WT - I_HL2A - I_HL2C - I_HL2B;
     END
    END
    ELSE
    BEGIN
      FP2 := 12;
```

```
      TFP2 := T1 - I_HL1C + WT - I_HL2A - I_HL2C;
    END
   END
   ELSE
   BEGIN
     FP2 := 13;
     TFP2 := T1 - I_HL1C + WT - I_HL2A;
   END
  END
  ELSE
  BEGIN
    FP2 := 11;
    TFP2 := T1 - I_HL1C + WT;
  END;
END;
IF T1+T2<=(I_HL2C - WT) THEN
BEGIN
  IF (T1+T2+I_HL3B)<=(I_HL2C - WT) THEN
  BEGIN
    FP3 := 18;
    TFP3 := T1+T2;
  END
  ELSE
  BEGIN
    FP3 := 17;
    TFP3 := I_HL3B + T1+T2 - I_HL2C + WT;
  END
END
ELSE
BEGIN
  IF (T1+T2 - I_HL2C+WT)>I_HL3A THEN
  BEGIN
   IF (T1+T2 - I_HL2C+WT - I_HL3A)>I_HL3C THEN
   BEGIN
    IF (T1+T2 - I_HL2C+WT - I_HL3A - I_HL3C)>I_HL3B THEN
    BEGIN
     IF (T1+T2-I_HL2C+WT-I_HL3A-I_HL3C-I_HL3B)>I_HL3A THEN
     BEGIN
      IF (T1+T2-I_HL2C+WT-2*I_HL3A-I_HL3C-I_HL3B)>I_HL3C THEN
      BEGIN
       IF (T1+T2-I_HL2C+WT-2*I_HL3A-2*I_HL3C-I_HL3B)>I_HL3B THEN
       BEGIN
        IF (T1+T2-I_HL2C+WT-2*I_HL3A-2*I_HL3C-2*I_HL3B)>I_HL3A
        THEN
        BEGIN
          FP3 := 18;
          TFP3 := T1+T2 - I_HL2C+WT-3*I_HL3A-2*I_HL3C-2*I_HL3B;
        END
        ELSE
        BEGIN
          FP3 := 16;
```

```
            TFP3 := T1+T2- I_HL2C+WT-2*I_HL3A-2*I_HL3C-2*I_HL3B;
          END
        END
        ELSE
        BEGIN
          FP3 := 17;
          TFP3 := T1+T2- I_HL2C+WT-2*I_HL3A-2*I_HL3C-I_HL3B;
        END
      END
      ELSE
      BEGIN
        FP3 := 18;
        TFP3 := T1+T2 - I_HL2C+WT-2*I_HL3A-I_HL3C-I_HL3B;
      END
    END
    ELSE
    BEGIN
      FP3 := 16;
      TFP3 := T1+T2 - I_HL2C + WT - I_HL3A - I_HL3C - I_HL3B;
    END
  END
  ELSE
  BEGIN
    FP3 := 17;
    TFP3 := T1+T2 - I_HL2C + WT - I_HL3A - I_HL3C;
  END
 END
 ELSE
 BEGIN
   FP3 := 18;
   TFP3 := T1+T2 - I_HL2C + WT - I_HL3A;
 END
END
ELSE
BEGIN
  FP3 := 16;
  TFP3 := T1+T2 - I_HL2C + WT;
END;
END;

AKHIR:      end;
      5 : begin
            status_2 := TRUE;
            exit;
          end;
      else begin
          status_2 := FALSE;
          exit;
      end;
    end;
    gotoxy(8,14); write('I_HL1A = ',I_HL1A:3);
    gotox
```

```
      y(8,15); write('I_HL1B = ',I_HL1B:3);
      gotoxy(8,16); write('I_HL1C = ',I_HL1C:3);
      gotoxy(30,14); write('I_HL1B = ',I_HL2A:3);
      gotoxy(30,15); write('I_HL2B = ',I_HL2B:3);
      gotoxy(30,16); write('I_HL3B = ',I_HL2C:3);
      gotoxy(52,14); write('I_HL1C = ',I_HL3A:3);
      gotoxy(52,15); write('I_HL2C = ',I_HL3B:3);
      gotoxy(52,16); write('I_HL3C = ',I_HL3C:3);
  until FALSE;
end;

Function Submenu_2 : byte;
var
   buffer_1 : bufferlayar;
   tekan    : char;
   pilihan  : byte;
begin
   box(6,7,75,17,$17);
   textattr := $1E;
   gotoxy(8,8);
   write('Lama lampu hijau perempatan I(A) menyala (detik)   :',
         II_HL1A:3);
   gotoxy(8,9);
   write('Lama lampu hijau perempatan I(B) menyala (detik)   :',
         II_HL1B:3);
   gotoxy(8,10);
   write('Lama lampu hijau perempatan II(B) menyala (detik)  :',
         II_HL2B:3);
   gotoxy(8,11);
   write('Lama lampu hijau perempatan III(B) menyala (detik) :',
         II_HL3B:3);
   gotoxy(8,12);
   write('Data jam sibuk sudah siap, kembali ke menu utama');
   gotoxy(8,14); write('II_HL1A = ',II_HL1A:3);
   gotoxy(8,15); write('II_HL1B = ',II_HL1B:3);
   gotoxy(8,16); write('II_HL1C = ',II_HL1C:3);
   gotoxy(30,14); write('II_HL2A = ',II_HL2A:3);
   gotoxy(30,15); write('II_HL2B = ',II_HL2B:3);
   gotoxy(30,16); write('II_HL2C = ',II_HL2C:3);
   gotoxy(52,14); write('II_HL3A = ',II_HL3A:3);
   gotoxy(52,15); write('II_HL3B = ',II_HL3B:3);
   gotoxy(52,16); write('II_HL3C = ',II_HL3C:3);

   Simpanlayar(6,7,75,17,buffer_1);
   textattr := $3F;
   pilihan := 1;
   rubahatribut(7,7+pilihan,74,7+pilihan,$3F);
   repeat
      tekan := readkey;
      case tekan of
        #0 : begin
```

```
                tekan := readkey;
                case tekan of
                   #80 : begin
                           inc(pilihan);
                           if pilihan > 5 then pilihan := 1;
                         end;
                   #72 : begin
                           dec(pilihan);
                           if pilihan < 1 then pilihan := 5;
                         end;
                end;
              end;
         #27: begin
                submenu_2 := 10;
                exit;
              end;
       end;
       kembalilayarsemula(6,7,75,17,buffer_1);
       rubahatribut(7,7+pilihan,74,7+pilihan,$3F)
    until tekan = #13;
    submenu_2 := pilihan;
end;

Procedure Sibuk;
LABEL
     ALFA, OMEGA;
var
  temp : byte;
begin
  status_3 := TRUE;
  repeat
     temp := SubMenu_2;
     case temp of
        1 : begin
              repeat
                gotoxy(68,8); II_HL1A := getb(3);
              until II_HL1A>0;
            end;
        2 : begin
                repeat
                   gotoxy(68,9); II_HL1B := getb(3);
                until II_HL1B > 0;
                   II_HL2A := II_HL1A + II_HL1B + WT;
            end;
        3 : begin
                repeat
                  gotoxy(68,10); II_HL2B := getb(3);
                until II_HL2B > 0;
                  II_HL3A := II_HL2A + II_HL2B + WT;
            end;
        4 : begin
```

```
                    repeat
                      gotoxy(68,11); II_HL3B := getb(3);
                    until II_HL3B > 0;
                      II_HL3C := II_HL3B - 5;
                      II_HL2C := II_HL3B + II_HL3C + WT;
                      II_HL1C := II_HL2B + II_HL2C + WT;

IF TJAM<7 THEN GOTO OMEGA;
IF TJAM<9 THEN GOTO ALFA;
IF TJAM<12 THEN GOTO OMEGA;
IF TJAM<14 THEN GOTO ALFA;
IF TJAM<16 THEN GOTO OMEGA;
IF TJAM<18 THEN GOTO ALFA;

ALFA:
IF T1<=(II_HL1C - WT) THEN
BEGIN
  IF (T1 + II_HL2B)<=(II_HL1C - WT) THEN
  BEGIN
    FP2 := 13;
    TFP2 := T1;
  END
  ELSE
  BEGIN
    FP2 := 12;
    TFP2 := II_HL2B + T1 - II_HL1C + WT;
  END
END
ELSE
BEGIN
  IF (T1 - II_HL1C+WT)>II_HL2A THEN
  BEGIN
   IF (T1 - II_HL1C+WT - II_HL2A)>II_HL2C THEN
   BEGIN
    IF (T1 - II_HL1C+WT - II_HL2A - II_HL2C)>II_HL2B THEN
    BEGIN
     IF (T1-II_HL1C+WT-II_HL2A-II_HL2C-II_HL2B)>II_HL2A THEN
     BEGIN
      IF (T1-II_HL1C+WT-2*II_HL2A-II_HL2C-II_HL2B)>II_HL2C THEN
      BEGIN
       IF (T1-II_HL1C+WT-2*II_HL2A-2*II_HL2C-II_HL2B)>II_HL2B
       THEN
       BEGIN
      IF (T1-II_HL1C+WT-2*II_HL2A-2*II_HL2C-2*II_HL2B)>II_HL2A
        THEN
        BEGIN
          FP2 := 13;
          TFP2 := T1-II_HL1C+WT-3*II_HL2A-2*II_HL2C-2*II_HL2B;
        END
        ELSE
        BEGIN
          FP2 :=
```

```
            11;
            TFP2 := T1-II_HL1C+WT-2*II_HL2A-2*II_HL2C-2*II_HL2B;
          END
        END
        ELSE
        BEGIN
          FP2 := 12;
          TFP2 := T1-II_HL1C+WT-2*II_HL2A-2*II_HL2C-II_HL2B;
        END
      END
      ELSE
      BEGIN
        FP2 := 13;
        TFP2 := T1 - II_HL1C+WT-2*II_HL2A-II_HL2C-II_HL2B;
      END
    END
    ELSE
    BEGIN
      FP2 := 11;
      TFP2 := T1 - II_HL1C + WT - II_HL2A - II_HL2C - II_HL2B;
    END
  END
  ELSE
  BEGIN
    FP2 := 12;
    TFP2 := T1 - II_HL1C + WT - II_HL2A - II_HL2C;
  END
 END
 ELSE
 BEGIN
   FP2 := 13;
   TFP2 := T1 - II_HL1C + WT - II_HL2A;
 END
END
ELSE
BEGIN
  FP2 := 11;
  TFP2 := T1 - II_HL1C + WT;
END;
END;
IF T1+T2<=(II_HL2C - WT) THEN
BEGIN
  IF (T1+T2+II_HL3B)<=(II_HL2C - WT) THEN
  BEGIN
    FP3 := 18;
    TFP3 := T1+T2;
  END
  ELSE
  BEGIN
    FP3 := 17;
    TFP3 := II_HL3B + T1+T2 - II_HL2C + WT;
  END
```

```
END
ELSE
BEGIN
  IF (T1+T2 - II_HL2C+WT)>II_HL3A THEN
  BEGIN
   IF (T1+T2 - II_HL2C+WT - II_HL3A)>II_HL3C THEN
   BEGIN
    IF (T1+T2 - II_HL2C+WT - II_HL3A - II_HL3C)>II_HL3B THEN
    BEGIN
     IF (T1+T2-II_HL2C+WT-II_HL3A-II_HL3C-II_HL3B)>II_HL3A THEN
     BEGIN
      IF (T1+T2-II_HL2C+WT-2*II_HL3A-II_HL3C-II_HL3B)>II_HL3C
      THEN
      BEGIN
       IF (T1+T2-II_HL2C+WT-2*II_HL3A-2*II_HL3C-II_HL3B)>II_HL3B
       THEN
       BEGIN
    IF (T1+T2-II_HL2C+WT-2*II_HL3A-2*II_HL3C-2*II_HL3B)>II_HL3A
        THEN
        BEGIN
        FP3 := 18;
        TFP3 := T1+T2-II_HL2C+WT-3*II_HL3A-2*II_HL3C-2*II_HL3B;
        END
        ELSE
        BEGIN
        FP3 := 16;
        TFP3 := T1+T2-II_HL2C+WT-2*II_HL3A-2*II_HL3C-2*II_HL3B;
        END
       END
       ELSE
       BEGIN
       FP3 := 17;
       TFP3 := T1+T2-II_HL2C+WT-2*II_HL3A-2*II_HL3C-II_HL3B;
       END
      END
      ELSE
      BEGIN
      FP3 := 18;
      TFP3 := T1+T2-II_HL2C+WT-2*II_HL3A-II_HL3C-II_HL3B;
      END
     END
     ELSE
     BEGIN
       FP3 := 16;
       TFP3 := T1+T2 -II_HL2C+WT-II_HL3A-II_HL3C-II_HL3B;
     END
    END
    ELSE
    BEGIN
      FP3 := 17;
```

```
        TFP3 := T1+T2 - II_HL2C + WT - II_HL3A - II_HL3C;
      END
     END
     ELSE
     BEGIN
       FP3 := 18;
       TFP3 := T1+T2 - II_HL2C + WT - II_HL3A;
     END
    END
    ELSE
    BEGIN
      FP3 := 16;
      TFP3 := T1+T2 - II_HL2C + WT;
    END;
 END;


 OMEGA:       end;
         5 : begin
                status_3 := TRUE;
                exit;
             end;
         else begin
             status_3 := FALSE;
             exit;
         end;
      end;
      gotoxy(8,14); write('II_HL1A = ',II_HL1A:3);
      gotoxy(8,15); write('II_HL1B = ',II_HL1B:3);
      gotoxy(8,16); write('II_HL1C = ',II_HL1C:3);
      gotoxy(30,14); write('II_HL2A = ',II_HL2A:3);
      gotoxy(30,15); write('II_HL2B = ',II_HL2B:3);
      gotoxy(30,16); write('II_HL2C = ',II_HL2C:3);
      gotoxy(52,14); write('II_HL3A = ',II_HL3A:3);
      gotoxy(52,15); write('II_HL3B = ',II_HL3B:3);
      gotoxy(52,16); write('II_HL3C = ',II_HL3C:3);
   until FALSE;
end;

Procedure Inisialisasi_RS232;
var
   r : registers;
begin
   r.AH := 0;
   r.AL := $C3;
   r.DX := PORT;
   Intr($14,r);
end;

Function RS232_siap : word;
var
```

```
    r : registers;
  begin
    r.DX := PORT;
    r.AH := 3;
    intr($14,r);
    RS232_siap := r.AX;
  end;

  Function Transfer(data : byte) : boolean;
  var
    r : registers;
  begin

    r.DX := PORT;
    r.AH := 1;
    r.AL := data;
    intr($14,r);
    if (r.AH and 128)=128 then Transfer := FALSE
    else Transfer := TRUE;
  end;

  Function Terima(var data: byte) : boolean;
  var
    r : registers;
    tekan : char;
  begin
    while (RS232_siap and 256)<>256 do
        if keypressed then
        begin
          tekan := readkey;
          terima := FALSE;
          exit;
        end;

    r.DX := PORT;
    r.AH := 2;
    intr($14,r);
    if (r.AH and 128)=128 then Terima := FALSE
    else
    begin
       data := r.AL;
       Terima := TRUE;
    end;
  end;

  Procedure SubMenu_3;
  var
     temp        : char;
     buff        : bufferlayar;
     ack         : byte;
  begin
```

```
If not(status_1) or not(status_2) or not(status_3) then
begin
   simpanlayar(30,20,50,22,buff);
   box(30,20,50,22,$4F);
   gotoxy(32,21); write('Data belum siap !');
   temp := readkey;
   kembalilayarsemula(30,20,50,22,buff);
   exit;
end;

Inisialisasi_RS232;

if not(Transfer(II_HL3C)) then
begin
   simpanlayar(30,20,50,22,buff);
   box(30,20,50,22,$4F);
   gotoxy(32,21); write('Transfer gagal !');
   temp := readkey;
   kembalilayarsemula(30,20,50,22,buff);
   exit;
end;

if not(Transfer(II_HL3B)) then
begin
   simpanlayar(30,20,50,22,buff);
   box(30,20,50,22,$4F);
   gotoxy(32,21); write('Transfer gagal !');
   temp := readkey;
   kembalilayarsemula(30,20,50,22,buff);
   exit;
end;

if not(Transfer(II_HL3A)) then
begin
   simpanlayar(30,20,50,22,buff);
   box(30,20,50,22,$4F);
   gotoxy(32,21); write('Transfer gagal !');
   temp := readkey;
   kembalilayarsemula(30,20,50,22,buff);
   exit;
end;

if not(Transfer(II_HL2C)) then
begin
   simpanlayar(30,20,50,22,buff);
   box(30,20,50,22,$4F);
   gotoxy(32,21); write('Transfer gagal !');
   temp := readkey;
   kembalilayarsemula(30,20,50,22,buff);
   exit;
end;
```

```
if not(Transfer(II_HL2B)) then
begin
   simpanlayar(30,20,50,22,buff);
   box(30,20,50,22,$4F);
   gotoxy(32,21); write('Transfer gagal !');
   temp := readkey;
   kembalilayarsemula(30,20,50,22,buff);
   exit;
end;

if not(Transfer(II_HL2A)) then
begin
   simpanlayar(30,20,50,22,buff);
   box(30,20,50,22,$4F);
   gotoxy(32,21); write('Transfer gagal !');
   temp := readkey;
   kembalilayarsemula(30,20,50,22,buff);
   exit;
end;

if not(Transfer(II_HL1C)) then
begin
   simpanlayar(30,20,50,22,buff);
   box(30,20,50,22,$4F);
   gotoxy(32,21); write('Transfer gagal !');
   temp := readkey;
   kembalilayarsemula(30,20,50,22,buff);
   exit;
end;

if not(Transfer(II_HL1B)) then
begin
   simpanlayar(30,20,50,22,buff);
   box(30,20,50,22,$4F);
   gotoxy(32,21); write('Transfer gagal !');
   temp := readkey;
   kembalilayarsemula(30,20,50,22,buff);
   exit;
end;

if not(Transfer(II_HL1A)) then
begin
   simpanlayar(30,20,50,22,buff);
   box(30,20,50,22,$4F);
   gotoxy(32,21); write('Transfer gagal !');
   temp := readkey;
   kembalilayarsemula(30,20,50,22,buff);
   exit;
end;

if not(Transfer(I_HL3C)) then
```

```
 begin
    simpanlayar(30,20,50,22,buff);
    box(30,20,50,22,$4F);
    gotoxy(32,21); write('Transfer gagal !');
    temp := readkey;
    kembalilayarsemula(30,20,50,22,buff);
    exit;
 end;

 if not(Transfer(I_HL3B)) then
 begin
    simpanlayar(30,20,50,22,buff);
    box(30,20,50,22,$4F);
    gotoxy(32,21); write('Transfer gagal !');
    temp := readkey;
    kembalilayarsemula(30,20,50,22,buff);
    exit;
 end;

 if not(Transfer(I_HL3A)) then
 begin
    simpanlayar(30,20,50,22,buff);
    box(30,20,50,22,$4F);
    gotoxy(32,21); write('Transfer gagal !');
    temp := readkey;
    kembalilayarsemula(30,20,50,22,buff);
    exit;
 end;

 if not(Transfer(I_HL2C)) then
 begin
    simpanlayar(30,20,50,22,buff);
    box(30,20,50,22,$4F);
    gotoxy(32,21); write('Transfer gagal !');
    temp := readkey;
    kembalilayarsemula(30,20,50,22,buff);
    exit;
 end;

 if not(Transfer(I_HL2B)) then
 begin
    simpanlayar(30,20,50,22,buff);
    box(30,20,50,22,$4F);
    gotoxy(32,21); write('Transfer gagal !');
    temp := readkey;
    kembalilayarsemula(30,20,50,22,buff);
    exit;
 end;

 if not(Transfer(I_HL2A)) then
 begin
```

```
    simpanlayar(30,20,50,22,buff);
    box(30,20,50,22,$4F);
    gotoxy(32,21); write('Transfer gagal !');
    temp := readkey;
    kembalilayarsemula(30,20,50,22,buff);
    exit;
end;

if not(Transfer(I_HL1C)) then
begin
    simpanlayar(30,20,50,22,buff);
    box(30,20,50,22,$4F);
    gotoxy(32,21); write('Transfer gagal !');
    temp := readkey;
    kembalilayarsemula(30,20,50,22,buff);
    exit;
end;

if not(Transfer(I_HL1B)) then
begin
    simpanlayar(30,20,50,22,buff);
    box(30,20,50,22,$4F);
    gotoxy(32,21); write('Transfer gagal !');
    temp := readkey;
    kembalilayarsemula(30,20,50,22,buff);
    exit;
end;

if not(Transfer(I_HL1A)) then
begin
    simpanlayar(30,20,50,22,buff);
    box(30,20,50,22,$4F);
    gotoxy(32,21); write('Transfer gagal !');
    temp := readkey;
    kembalilayarsemula(30,20,50,22,buff);
    exit;
end;

if not(Transfer(Tdtk)) then
begin
    simpanlayar(30,20,50,22,buff);
    box(30,20,50,22,$4F);
    gotoxy(32,21); write('Transfer gagal !');
    temp := readkey;
    kembalilayarsemula(30,20,50,22,buff);
    exit;
end;

if not(Transfer(Tmnt)) then
begin
    simpanlayar(30,20,50,22,buff);
```

```
    box(30,20,50,22,$4F);
    gotoxy(32,21); write('Transfer gagal !');
    temp := readkey;
    kembalilayarsemula(30,20,50,22,buff);
    exit;
end;

if not(Transfer(Tjam)) then
begin
    simpanlayar(30,20,50,22,buff);
    box(30,20,50,22,$4F);
    gotoxy(32,21); write('Transfer gagal !');
    temp := readkey;
    kembalilayarsemula(30,20,50,22,buff);
    exit;
end;

if not(Transfer(FP2)) then
begin
    simpanlayar(30,20,50,22,buff);
    box(30,20,50,22,$4F);
    gotoxy(32,21); write('Transfer gagal !');
    temp := readkey;
    kembalilayarsemula(30,20,50,22,buff);
    exit;
end;

if not(Transfer(TFP2)) then
begin
    simpanlayar(30,20,50,22,buff);
    box(30,20,50,22,$4F);
    gotoxy(32,21); write('Transfer gagal !');
    temp := readkey;
    kembalilayarsemula(30,20,50,22,buff);
    exit;
end;

if not(Transfer(FP3)) then
begin
    simpanlayar(30,20,50,22,buff);
    box(30,20,50,22,$4F);
    gotoxy(32,21); write('Transfer gagal !');
    temp := readkey;
    kembalilayarsemula(30,20,50,22,buff);
    exit;
end;

if not(Transfer(TFP3)) then
begin
    simpanlayar(30,20,50,22,buff);
    box(30,20,50,22,$4F);
```

```
        gotoxy(32,21); write('Transfer gagal !');
        temp := readkey;
        kembalilayarsemula(30,20,50,22,buff);
        exit;
    end;

    if not(Transfer(PK)) then
    begin
        simpanlayar(30,20,50,22,buff);
        box(30,20,50,22,$4F);
        gotoxy(32,21); write('Transfer gagal !');
        temp := readkey;
        kembalilayarsemula(30,20,50,22,buff);
        exit;
    end;

end;

var
    buffer_1,buffer_2 : bufferlayar;
    temp : byte;
begin
    simpanlayar(1,1,80,25,buffer_1);
    I_HL1A := 0; I_HL1B := 0; I_HL1C := 0; I_HL2A := 0; PK:=0;
    I_HL2B := 0; I_HL2C := 0; I_HL3A := 0; I_HL3B := 0;
    I_HL3C := 0; II_HL1A := 0; II_HL1B := 0; II_HL1C := 0;
    II_HL2A := 0; II_HL2B := 0; II_HL2C := 0; II_HL3A := 0;
    II_HL3B := 0; II_HL3C := 0;
    textattr := $07;
    clrscr;
    repeat
        temp := Menu;
        case temp of
          1 : begin
                Simpanlayar(12,5,68,16,buffer_2);
                Inisialisasi;
                kembalilayarsemula(12,5,68,16,buffer_2);
              end;
          2 : begin
                Simpanlayar(6,7,75,17,buffer_2);
                Biasa;
                kembalilayarsemula(6,7,75,17,buffer_2);
              end;
          3 : begin
                Simpanlayar(6,7,75,17,buffer_2);
                Sibuk;
                kembalilayarsemula(6,7,75,17,buffer_2);
              end;
          4 : SubMenu_3;
          5 : begin
                kembalilayarsemula(1,1,80,25,buffer_1);
                exit;
              end;
```

```
        else begin
            kembalilayarsemula(1,1,80,25,buffer_1);
            exit;
        end;
      end;
   until FALSE;
end.
```

RIWAYAT HIDUP

Penulis dilahirkan di AMBON pada tanggal 10 Mei 1969, dari Ayah Phoa Njan Kie dan Ibu Wong Lie Koen.

Penulis terdaftar sebagai mahasiswa Institut Sepuluh Nopember di Surabaya pada Fakultas Teknologi Industri, Jurusan Teknik Elektro, Bidang Studi Elektronika, dengan Nomor Registrasi 2882200986. Selama menjadi mahasiswa pernah menjadi asisten praktikum pada laboratorium Mikroelektronika, laboratorium Elektronika, dan laboratorium Rangkaian Listrik periode 1992/1993.

Jenjang pendidikan yang telah ditempuh oleh penulis sebelumnya adalah sebagai berikut:

- SD Xaverius A1 AMBON, lulus tahun 1982
- SMP Xaverius AMBON, lulus tahun 1985
- SMAK Kolese Santo Yusup MALANG, lulus tahun 1988